

AD-A033 948

STANFORD UNIV CALIF DEPT OF OPERATIONS RESEARCH  
COMPUTER PROGRAMS FOR DECOMPOSITION IN INTEGER PROGRAMMING.(U)  
SEP 76 G A KOCHMAN  
TR-71

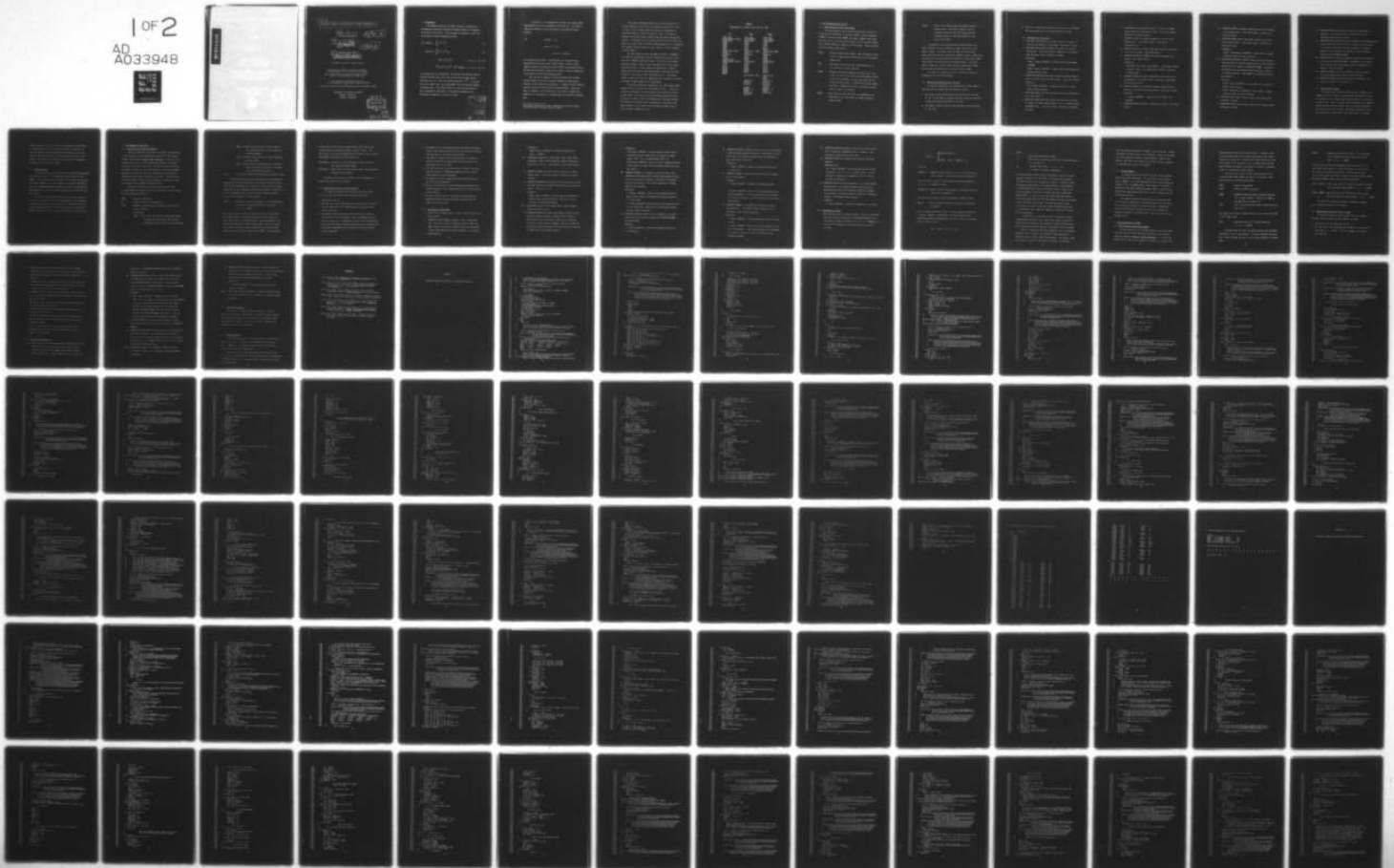
F/G 9/2

N00014-76-C-0418

NL

UNCLASSIFIED

1 OF 2  
AD  
A033948



6

COMPUTER PROGRAMS FOR DECOMPOSITION IN INTEGER PROGRAMMING •

10

by

Gary A. Kochman

14 TR-71

9

TECHNICAL REPORT, NO. 71

11

September 1976

12 167p.

15

PREPARED UNDER CONTRACT

NO0014-76-C-0418 (NR-047-061)

DAAC29-74-C-0034  
FOR THE OFFICE OF NAVAL RESEARCH

Frederick S. Hillier, Project Director

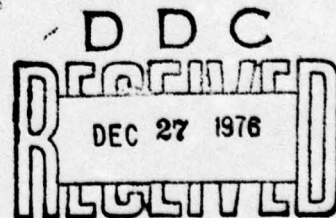
Reproduction in Whole or in Part is Permitted  
for any Purpose of the United States Government

This document has been approved for public release  
and sale; its distribution is unlimited.

This research was supported in part by  
U.S. Army Research Office Contract DAAC29-74-C-0034  
and  
Air Force Office of Scientific Research Contract F44620-74-C-0079

DEPARTMENT OF OPERATIONS RESEARCH ✓

STANFORD UNIVERSITY  
STANFORD, CALIFORNIA



1473  
402766 LB



## 1. Introduction

The computer codes DSLC and DMLC, written to implement the decomposition algorithms developed by Kochman (1976), are documented and listed in this report. These programs are designed to solve the block angular integer programming problem,

$$(P) \text{ maximize } \sum_{j=1}^N c^{(j)} x^{(j)}, \quad (1)$$

$$\text{subject to } \sum_{j=1}^N A^{(j)} x^{(j)} \leq b \quad (2)$$

$$B^{(j)} x^{(j)} \leq \beta^{(j)} \quad j = 1, 2, \dots, N \quad (3)$$

$$L^{(j)} \leq x^{(j)} \leq U^{(j)}, \quad x^{(j)} \text{ integer}, \quad j = 1, 2, \dots, N, \quad (4)$$

by decomposition into subproblems. The routine DSLC exploits certain special structures which arise when (P) has only a single linking constraint (2), i.e., when the constraint matrices  $A^{(j)}$  ( $1 \leq j \leq N$ ) have only a single row. The routine DMLC treats the multiple linking constraints case. See Kochman (1976) for a detailed discussion of these decomposition algorithms; a step-by-step statement of the algorithms is presented in Section 5.2 there.

ACCESSION FOR	
NTIS	Write Section <input checked="" type="checkbox"/>
DOC	Buff Section <input type="checkbox"/>
UNCLASSIFIED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. and/or SPECIAL
A	

In addition to the programs DSLC and DMLC, the integer linear programming routine BB is documented and listed here. The computer code BB is designed to solve the general (unstructured) integer program,

$$\begin{aligned} \text{(Q)} \quad & \text{maximize} \quad cx \\ & \text{subject to} \quad Ax \leq b \\ & L \leq x \leq U, \quad x \text{ integer,} \end{aligned}$$

by branch-and-bound search. Specifically, the branch-and-bound algorithm BB follows the ideas of Dakin (1965). Tomlin's (1970) improved penalty calculations are used to guide the branching process, and the last-in-first-out (LIFO) rule is used to retrieve stored nodes in the search tree when backtracking occurs<sup>1</sup>.

The code BB is embedded in both the routines DSLC and DMLC for the purpose of solving the integer programming subproblems as required by the decomposition procedures. It should be noted, however, that the routine BB itself uses only standard methods. This code does not represent a new contribution to the field of integer programming; its documentation is included here for the sake of completeness.

---

<sup>1</sup>For further references, see Beale (1968), Beale and Small (1965), and Little, Murty, Sweeney, and Karel (1963).



The linear programming portions of all three programs are based extensively on the Fortran code LPM-1, developed by Dr. John A. Tomlin of the Systems Optimization Laboratory, Department of Operations Research, Stanford University. LPM-1 is designed to implement the revised simplex method, using the product-form of the inverse. In adopting this code for use in the routines BB, DSLC, and DMLC, the present author has modified LPM-1 appropriately to incorporate the bounded-variable technique as well. LPM-1 has been documented by Tomlin (1975).

The three codes BB, DSLC, and DMLC are written in FORTRAN IV for Stanford's IBM 370/168, with all instructions and data in-core. Each code is WATFIV compatible as well. The codes may be machine-dependent, however, insofar as each code makes calls to the system subroutine PCLOCK. In each program, this subroutine is used to compute the total time the IBM 370/168 Central Processing Unit was available to the core partition containing the job. In addition, PCLOCK is used to determine the computation times required for various intermediate tasks performed by the algorithms.

The various subprograms (exclusive of the main programs) which comprise each of the three codes are listed in Table I. These subroutines are described in greater detail in the remaining sections of this report. More specifically, the branch-and-bound code BB is discussed in Section 2, while the decomposition codes DSLC and DMLC are discussed in Sections 3 and 4, respectively. Listings of the three programs, and the output results obtained in sample runs of each are given in Appendices I-III.



TABLE I

Subprograms for Computer Codes BB, DSLC, DMLC

<u>BB</u>	<u>DSLC</u>	<u>DMLC</u>
BLOCK DATA	BLOCK DATA	BLOCK DATA
INPUT(IFPROB, INITBD)	INPUT(IFPROB)	INPUT(IFPROB)
FTRAN(IPAR)	FTRAN(IPAR)	FTRAN(IPAR)
BTRAN	BTRAN	BTRAN
FORMC	FORMC	FORMC
PRICE	PRICE	PRICE
CHUZR	CHUZR	CHUZR
WRETA	WRETA	WRETA
SHIFTR(IOLD, INEW)	SHIFTR(IOLD, INEW)	SHIFTR(IOLD, INEW)
INVERT	INVERT	INVERT
UNPACK(IV)	UNPACK(IV)	UNPACK(IV)
SHFTE	SHFTE	SHFTE
UPBETA	UPBETA	UPBETA
NORMAL(ITSINV)	NORMAL	NORMAL
BANDB(INITBD, ITSINV)	BANDB(INITBD)	BANDB(INITBD)
DCHUZR	DCHUZR	DCHUZR
DCHUZO	DCHUZO	DCHUZO
TESTX	TESTX	TESTX
PENLTS	PENLTS	PENLTS
BRANCH	BRANCH	BRANCH
BKTRAK	BKTRAK	BKTRAK
WRAPUP		
	ALOCTE(IPAR, IRHS)	ALOCTE(IPAR)
		LDMSTR
		TESTZ
	DOMCHK(J, K)	DOMCHK(J, K)
	LDDATA(J)	LDDATA(J)
	GETLBD(J)	GETLBD(J)
		PISAVR(J)
	SAVER(J)	SAVER(J)
	DMBRAN(J, K)	DMBRAN(J, K)
	BBKTRK	BBKTRK
	OUTSOL(INC)	OUTSOL(INC)
	PHASE1(J)	

## 2. The Branch-and-Bound Code BB

### 2.1. Main Program and Input Requirements

The main program for the branch-and-bound code BB serves to coordinate the application of the input (INPUT), linear programming (NORMAL), branch-and-bound (BANDB), and output (WRAPUP) subroutines. The first card of input must contain values for the variables INPROB, IOBJ, INVFRQ, ITRFRQ, and INITBD in 4I4, I10 format. These variables have the following significance:

- INPROB        = problem identification number. (Must be nonzero; otherwise, no computations are carried out and BB terminates immediately.)
- IOBJ           = row number for objective row. Automatically set to one by default if IOBJ is omitted.
- INVFRQ        = Parameter determining frequency with which basis reinversion is carried out in the linear programming portions of the code; if INVFRQ = k, basis reinversion will be carried out after every k (primal and/or dual) simplex iterations. INVFRQ should not be set greater than the number of rows in the problem.
- ITRFRQ        = upper limit on total iterations; all computations are terminated if the total number of simplex iterations exceeds ITRFRQ.



INITBD = initial lower bound estimate for maximum objective function value. The value of INITBD is used in fathoming tests until the first integer-feasible incumbent solution is found with objective value greater than INITBD.

BB assumes all cost coefficients are integer-valued, and takes advantage of this by rounding all upper bounds down to the nearest integer. The formats used for input of row and column names, objective function coefficients, constraint matrix coefficients, and right-hand side coefficients are identical to those used by LPM-1. (This format is consistent with standard MPS format.) Following input of these data, initial lower and upper bounds XLB(J), XUB(J) for all decision variables must be input in 15F5.0 format.

The input for a sample run of BB is included in Appendix I, following the listing of the program.

## 2.2. Restrictions Relevant to the Use of BB

As currently configured, the following restrictions apply to the input data and problem size and complexity for BB:

1. The number of rows (including the objective function row) must be .LE. 60; the number of columns (including slack and artificial variables generated by BB) must be .LE. 122.
2. The number of nonzero objective and constraint coefficients must be .LE. 2000.



3. The list of stored nodes generated in the course of the branch-and-bound search must never exceed 500 nodes in length.

### 2.3. Description of Subprograms

1. BLOCK DATA--From LPM-1; sets initial values for a number of variables used throughout the program. (See Tomlin (1975).)
2. SUBROUTINE INPUT(IFPROB, INITBD)--From LPM-1; reads in all problem data. (See Tomlin (1975).) Modified to also read in lower and upper bounds on decision variables.

Parameters:

IFPROB (Output, INTEGER\*4) Problem identification number  
(=INPROB)

INITBD (Output, INTEGER\*4) Initial lower bound estimate for  
maximum objective value.

3. SUBROUTINE FTRAN(IPAR)--From LPM-1; performs forward transformation of matrix columns by basis inverse. (See Tomlin (1975).)

Parameters:

IPAR (Input, INTEGER\*4) Determines eta-vectors by which  
matrix column is updated.

4. SUBROUTINE BTRAN--From LPM-1; performs backward transformation.  
(See Tomlin (1975).)
5. SUBROUTINE FORMC--From LPM-1; forms objective function vector for Phase I of primal simplex method. Used in conjunction with SUBROUTINE PRICE. (See Tomlin (1975).) Modified for bounded-variables.

6. SUBROUTINE PRICE--From LPM-1; prices out columns for primal simplex method and chooses pivot column. (See Tomlin (1975).) Modified for bounded-variables.
7. SUBROUTINE CHUZR--From LPM-1; chooses pivot row for primal simplex method. (See Tomlin (1975).) Modified for bounded-variables.
8. SUBROUTINE WRETA--From LPM-1; forms new eta-vectors for product-form of basis inverse. (See Tomlin (1975).)
9. SUBROUTINE SHIFTR(IOLD, INEW)--From LPM-1; rearranges data storage. (See Tomlin (1975).)

Parameters:

IOLD, INEW (Both input, both INTEGER\*4) IOLD indexes storage locations of data to be transferred by SHIFTR to locations indexed by INEW.

10. SUBROUTINE INVERT--From LPM-1; Determines basis inverse by LU decomposition. (See Tomlin (1975).) Modified to update RHS with bounded-variables.
11. SUBROUTINE UNPACK(IV)--From LPM-1; expands compressed matrix columns by inserting zeroes appropriately. (See Tomlin (1975).)

Parameters:

IV (Input, INTEGER\*4) Indexes the matrix column to be expanded.

12. SUBROUTINE SHFTE--From LPM-1; subroutine for INVERT. (See Tomlin (1975).)



13. SUBROUTINE UPBETA--From LPM-1; updates RHS following a primal or dual simplex pivot. (See Tomlin (1975).) Modified for bounded-variables.

14. SUBROUTINE NORMAL(ITSINV)--From LPM-1; directs execution of primal simplex method. (See Tomlin (1975).) Modified for bounded-variables.

Parameters:

ITSINV (Input/Output, INTEGER\*4) Counts number of simplex iterations since last basis inversion.

15. SUBROUTINE BANDB(INITBD, ITSINV)--directs execution of branch-and-bound search for optimal integer solution, after optimal LP solution has been computed in SUBROUTINE NORMAL. Also directs reoptimization via dual simplex method following a branch in the forward direction. Calls NORMAL for reoptimization after backtracking.

Parameters:

INITBD (Input, INTEGER\*4) Initial lower bound estimate for maximal objective value.

ITSINV (Input/Output, INTEGER\*4) Counts number of simplex iterations since last basis inversion.

16. SUBROUTINE DCHUZR--Selects pivot row for dual simplex method (analogous to CHUZR).

17. SUBROUTINE DCHUZR--Selects pivot column for dual simplex method (Analogous to PRICE).



18. SUBROUTINE TESTX--Tests optimal LP solution at each node for fathoming. Installs new incumbent each time an improved integer-feasible solution is found.
19. SUBROUTINE PENLTS--Computes Tomlin's improved up- and down-penalties and the Gomory penalty at each node. Checks for forced branches on both basic and non-basic decision variables. If computed penalties are insufficient for fathoming, chooses branching variable, calls SUBROUTINE BRANCH.
20. SUBROUTINE BRANCH--Performs necessary bookkeeping for branching as indicated by SUBROUTINE PENLTS. Increments list of stored nodes; revises bounds on branching variable appropriately for branch in forward direction.
21. SUBROUTINE BKTRAK--Uses last-in-first-out (LIFO) strategy to retrieve unfathomed nodes from list of stored nodes.
22. SUBROUTINE WRAPUP--Outputs final solution information. (See output from sample run in Appendix I.)

#### 2.4. Description of Output

The output from BB takes the following form. Whenever a new incumbent solution is found, the computation time required to reach the solution, and the corresponding objective value of the solution are printed out. When all computations are completed (i.e., when the optimal solution has been obtained, and its optimality verified), the total solution time is output. The optimal integer solution is then printed out in 2015 format. More specifically, the optimal

column values are output in the order (1) objective function value, (2) slack variables, and (3) decision variables. The values of the first twenty columns are printed on the first line, the second twenty on the second line, etc., until all column values have been printed. The output from a sample run of BB is given in Appendix I, following the listing of the program.

#### 2.5. Further Comments

As currently written, the Fortran code BB contains approximately 1,450 source statements. All significant array storage is done in COMMON. In all, approximately 64K bytes of core storage are required by the three COMMON blocks. When compiled on Stanford's IBM 370/168 under IBM's FORTRAN-H-extended compiler, with level two optimization, approximately 113K bytes of core are required for all instruction and data storage.

The computer program BB was written by the present author in the course of completing his doctoral dissertation in the Department of Operations Research, Stanford University. Computational results (solution times) obtained on several test problems are presented in Chapter 6 of Kochman (1976). BB was last revised in September 1976.

### 3. The Decomposition Code DSLC

#### 3.1. Main Program and Input Requirements

The main program for the code DSLC coordinates the application of the two phases of the decomposition algorithm for block angular integer programs with a single linking constraint. In Phase I, the problem data is input, and the computations required to construct the Master Problem are carried out. The branch-and-bound search for an optimal allocation of the linking resources is executed in Phase 2. DSLC has been written so that any of the four decision rules for subproblem selection may be specified by the user. (See Section 5.2 in Kochman (1976).) This is done by assigning the appropriate value to the input variable IPARM, as described below.

The first card of input for DSLC contains values for the variables NSUBS, IB, IPARM, and IZINC in 4I10 format. These variables have the following significance:

NSUBS        = number of subproblems.

IB            = linking constraint right-hand side.

IPARM        = parameter determining which subproblem selection rule is used in Phase 2.

Specifically,

IPARM = 1  $\Rightarrow$  Selection rule based on average gaps between  
LP bounds and corresponding exact integer  
programming objective values in the subproblems.



IPARM = 2  $\Rightarrow$  Selection rule based on average slack in  
the linking constraint in the optimal sub-  
problem solutions.

IPARM = 3  $\Rightarrow$  Selection rule based on average subproblem  
solution times.

IPARM = 4  $\Rightarrow$  First unfixed subproblem will be selected,  
where subproblems are numbered in the same  
order in which they were input.

IZINC = lower bound estimate for maximum overall objective value.

Following the first card, the problem data for the individual subproblems is read in sequentially. The format of this data is the same for each subproblem, and is similar to the format used to input problem data in the branch-and-bound code BB described in Section 2.1. For each subproblem, the first card of input contains the value for the variable INPROB in I4 format, where

INPROB = subproblem identification number. The subproblems must be numbered and input sequentially in the order 1, 2, 3, ..., NSUBS.

DSLIC assumes that all cost coefficients in each of the subproblems are integer-valued, and takes advantage of this by rounding all upper bounds down to the nearest integer. For each subproblem, the formats used for input of row and column names, objective function coefficients, constraint matrix coefficients, and right-hand side coefficients are identical to those used by LPM-1. Following input

of these data, initial lower and upper bounds  $XLB(J)$ ,  $XUB(J)$  must be specified for the decision variables in 15F5.0 format. The final input card for each subproblem contains values for the variables  $IBLB(INPROB)$ ,  $IBUB(INPROB)$  in 2I10 format, where

$IBLB(INPROB)$  = lower bound on the allocation variable associated with subproblem number  $INPROB$ .

$IBUB(INPROB)$  = upper bound on the allocation variable associated with subproblem number  $INPROB$ .

The input for a sample run of DSLC is given in Appendix II, following the listing of the program.

### 3.2. Restrictions Relevant to the Use of DSLC

As currently configured, the following restrictions apply to the input data and problem size and complexity for DSLC:

1. NSUBS must be .LE. 10.
2. The first row in each subproblem must correspond to the objective row; the second row in each subproblem must correspond to the linking constraint.
3. No right-hand side value should be specified for the linking constraint row in each subproblem. Prior to computing the parametric LP solution for a subproblem in Phase 1, DSLC automatically initializes the linking constraint right-hand side to the specified lower bound  $IBLB(INPROB)$  on the allocation variable associated with the subproblem.



4. The number of rows (including objective and linking constraint rows) in each subproblem must be .LE. 22; the number of decision variables must be .LE. 50 in each subproblem.
5. The number of nonzero objective and constraint coefficients must be .LE. 500 in each subproblem; the total number in all subproblems must be .LE. 1000.
6. The modified concave and piecewise linear parametric objective function generated in SUBROUTINE PHASE1(J) must have .LE. 50 linear segments in each subproblem.
7. The number of distinct subproblem solutions generated must be .LE. 200 for each subproblem.
8. The list of stored nodes in the branch-and-bound search for an optimal allocation of the linking resource must never exceed 200 nodes in length.
9. The list of stored nodes in the branch-and-bound search for optimal subproblem solutions must never exceed 500 nodes in length for any subproblem.

### 3.3. Description of Subprograms

1. BLOCK DATA--From LPM-1; same as for BB. (See Section 2.3 and Tomlin (1975).)
2. SUBROUTINE INPUT(IFPROB)--From LPM-1; reads in subproblem data. (See Tomlin (1975).) Modified to read in lower and upper bounds on the decision variables and on the allocation variable associated with the given subproblem. Also stores subproblem data.

Parameters:

IFPROB (Output, INTEGER\*4) Subproblem identification number (= INPROB).

- 3-13. SUBROUTINES FTRAN(IPAR), BTRAN, FORMC, PRICE, CHUZR, WRETA, SHIFTR(IOLD, INEW), INVERT, UNPACK(IV), SHFTE, UPBETA--From LPM-1; same as for BB. (See Section 2.3, and Tomlin (1975).)
14. SUBROUTINE NORMAL--From LPM-1; directs execution of primal simplex method. (See Tomlin (1975).) Modified for bounded-variables.
15. SUBROUTINE BANDB(INITBD)--From BB; directs execution of branch-and-bound search for an optimal integer solution to given subproblem. (See Section 2.3.)

Parameters:

INITBD (Input, INTEGER\*4) Initial lower bound estimate for maximum subproblem objective value. Value of INITBD is specified by SUBROUTINE GETLBD(J).

- 16-21. SUBROUTINES DCHUZR, DCHUZO, TESTX, PENLTS, BRANCH, BKTRAK--From BB (See Section 2.3).
22. SUBROUTINE ALOCTE(IPAR, IRHS)--Solves Master Problem (concave-separable integer program; see Kochman (1976), Chapter 4). When current node in allocation-variables search tree was obtained by forward branch, the Master Problem is reoptimized by starting from the previously optimal solution and reallocating IRHS units of the linking resource.



Parameters:

IPAR (Input, INTEGER\*4) Indicates whether current node in allocation-variables search tree was obtained by forward branch (IPAR = 1), or by backtracking (IPAR = 0).

IRHS (Input, INTEGER\*4) Amount of slack in linking constraint in the optimal solution to last subproblem solved; used only when IPAR = 1.

23. SUBROUTINE DOMCHK(J, K)--Compares current allocation (from SUBROUTINE ALOCTE(IPAR, IRHS)) with previously examined allocations in each unfixed subproblem to determine if the optimal solution corresponding to the current allocation is already known for any subproblem.

Parameters:

J (Output, INTEGER\*4) Indexes first subproblem found for which optimal solution corresponding to current allocation is already known.

K (Output, INTEGER\*4) Indexes the previously examined allocation in subproblem J which leads to same optimal solution as would current allocation.

24. SUBROUTINE LDDATA(J)--Loads data for subproblem J into appropriate storage locations for solution by SUBROUTINES NORMAL and BANDB(INITBD).

Parameters:

J (Input, INTEGER\*4) Indexes the subproblem selected to be solved next.

25. SUBROUTINE GETLBD(J)--Compares current allocation in subproblem J with previously examined allocations to determine an initial lower bound on the maximal objective value in subproblem J under the current allocation.

Parameters:

J (Input, INTEGER\*4) Indexes the subproblem selected to be solved next.

26. SUBROUTINE SAVER(J)--Stores allocation and solution information for subproblem J.

Parameters:

J (Input, INTEGER\*4) Indexes last subproblem solved.

K (Input, INTEGER\*4) Indexes a previously examined allocation in subproblem J. The current allocation in subproblem J is fixed at the level of the  $K^{\text{th}}$  allocation by the branching procedure.

27. SUBROUTINE DMBRAN(J, K)--Branches on allocation variables in subproblem J. (See Kochman (1976), Chapter 3 for a discussion of the branching procedure.)

Parameters:

J (Input, INTEGER\*4) Indexes subproblem in which branching is done.

K (Input, INTEGER\*4) Indexes a previously examined allocation in subproblem J. The current allocation in subproblem J is fixed at the level of the K allocation by the branching procedure.



28. SUBROUTINE BBKTRK--Backtracks in allocation-variables search tree to locate next unfathomed node to be examined. Last-in-first-out (LIFO) selection rule used.

29. SUBROUTINE OUTSOL(INC)--Outputs final solution information.

Parameters:

DIMENSION INC(10)

INC (Input, INTEGER\*4) Array containing optimal solution information. For  $J = 1, 2, \dots, \text{NSUBS}$ ,  $\text{INC}(J) = k$  if the solution in subproblem  $J$  corresponds to the  $k^{\text{th}}$  solution obtained in subproblem  $J$ .

30. SUBROUTINE PHASE1(J)--Solves subproblem  $J$  as a linear program parametrically in the linking constraint right-hand side. Simultaneously modifies the resulting concave and piecewise linear parametric objective function to render all breakpoints integral. (See Kochman (1976), Chapter 4.)

Parameters:

J (Input, INTEGER\*4) Indexes next subproblem to be solved.

### 3.4. Description of Output

The following output is generated by DSLC. After each subproblem  $J$  has been solved parametrically in SUBROUTINE PHASE1(J), the (all-integer) breakpoints  $\text{IBP}(K, J)$  ( $K = 1, 2, \dots, \text{NIAMDA}(J) + 1$ ) of the resulting parametric objective function are printed out, where

$$IBP(K, J) = \begin{cases} IBLB(J), & \text{when } K = 1 \\ IBUB(J), & \text{when } K = NLAMDA(J) + 1, \end{cases}$$

and where

$NLAMDA(J)$  = number of linear segments in the concave and piecewise linear parametric objective function for subproblem J.

The second line of output for each subproblem displays  $RLAMADA(K, J)$  ( $K = 1, 2, \dots, NLAMDA(J)$ ), where

$RLAMADA(K, J)$  = slope of the  $K^{\text{th}}$  linear segment of parametric objective functions for subproblem J.

The third line of output for each subproblem gives  $ZNAUT(J)$ , where

$ZNAUT(J)$  = value of parametric objective function for subproblem J at  $IBLB(J)$ .

Next, the output generated in Phase 2 of the decomposition algorithm is printed. Whenever a subproblem is (re)solved explicitly by SUBROUTINES NORMAL and BANDB(INITBD), one line of output is generated; this line takes the form,

SUB J, SOL K:  $X_1 \ X_2, \dots, X_n$ ,



where

- J           = subproblem identification number
- K           = identification number associated with the given solutions  
            for subproblem J.
- $X_l$        = optimal column values,  $l = 1, 2, \dots, n$ ,  
            n = number of columns in subproblem J.

As in the branch-and-bound code BB, the optimal solution to the subproblem is printed out in the order (1) objective function value, (2) slack variables, and (3) decision variables. It should be noted that when the lower bound on the maximal objective value generated in SUBROUTINE GETLBD(J) equals the optimal objective value for the given subproblem, no incumbent solutions are found in SUBROUTINE BANDB(INITBD). In this case, only the optimal objective function value and linking constraint slack are printed out; zeroes are printed for the remaining column values. The remaining optimal column values may be recovered through the solution identification number K, by referring to the output for subproblem J when the  $K^{\text{th}}$  solution was first generated.

In addition to the subproblem solution information, whenever a new incumbent solution to the overall block angular problem (P) is discovered, the computation time required to reach the solution and the corresponding incumbent objective value IZINC in (P) are printed.

When all computations have been completed, the overall optimal solution to (P) is printed. The solution in each subproblem is identified by the solution identification number. The optimal column values may be recovered by referring to the output of subproblem

solution information generated in Phase 2 of the algorithm. Finally, the maximal objective value to (P) is printed, followed by the output of various parameters related to the performance of DSLC on the problem. The output from a sample run of DSLC is given in Appendix II, following the listing of the program.

### 3.5. Further Comments

As currently written the Fortran code DSLC contains approximately 1,950 source statements. All significant array storage is done in COMMON. In all, approximately 65K bytes of core storage are required by the five COMMON blocks. When compiled on Stanford's IBM 370/168 under IBM's FORTRAN-H-Extended compiler, with level two optimization, approximately 126K bytes of core are required for all instructions and data.

The computer program DSLC was written by the present author in the course of completing his doctoral dissertation in the Department of Operations Research, Stanford University. Computational results (solution times) obtained on several test problems are presented in Chapter 6 of Kochman (1976). DSLC was last revised in September 1976.

## 4. The Decomposition Code DMLC

### 4.1. Main Program and Input Requirements

The main program for the code DMLC coordinates the application of the two phases of the decomposition algorithm for block angular integer programs with multiple linking constraints. As in DSLC, the problem data is input and the computations required to construct the



Relaxed Master Problem are executed in Phase 1. In Phase 2, the branch-and-bound search for an optimal allocation of the linking resources is carried out. The major differences between the codes DSLC and DMLC are in the methods used to construct and solve the Master Problem; see Chapter 4 and Chapter 5, Section 1 in Kochman (1976) for a discussion of these methods.

The first card of input for DMLC contains values for the variables NSUBS, NLINK, IPARM, and IZINC in 4I10 format. These variables have the following significance:

NSUBS	= number of subproblems
NLINK	= number of linking constraints
IPARM	= parameter determining which subproblem selection rule is used in Phase 2. Identical to IPARM in the code DSLC; see Section 3.1.
IZINC	= lower bound estimate for maximum overall objective value in (P).

The second card of input for DMLC contains values for the array IB(I) in 5I10 format, where

IB(I)	= right-hand side for $I^{\text{th}}$ linking constraint,
	$I = 1, 2, \dots, \text{NLINK}.$

Following these two cards, the problem data for the individual subproblems is read in sequentially. For each subproblem, the first card of input contains the value for the variable INPROB in I4 format, where

INPROB            = subproblem identification number. The subproblems must be numbered and input sequentially in the order 1, 2, 3, ..., NSUBS.

The format for the individual subproblem data is identical to that used in DSLC (see Section 3.1), with the exception of the input of the bounds on the allocation variables. The final two cards of input for each subproblem contains values for the arrays IBLB(I, INPROB) and IBUB(I, INPROB), respectively, in 5I10 format, where

IBLB(I, INPROB) = lower bound on the  $I^{\text{th}}$  allocation variable associated with subproblem number INPROB,  $I = 1, 2, \dots, \text{NLINK}$ .

IBUB(I, INPROB) = upper bound on the  $I^{\text{th}}$  allocation variable associated with subproblem number INPROB,  $I = 1, 2, \dots, \text{NLINK}$ .

The input for a sample run of DMLC is displayed in Appendix III, following the listing of the program.

#### 4.2. Restrictions Relevant to the Use of DMLC

As currently configured, the following restrictions apply to the input data and problem size and complexity for DMLC:

1. NSUBS must be .LE. 10; NLINK must be .LE. 5.
2. The first row in each subproblem must correspond to the objective row; rows 2, 3, ..., NLINK + 1 must correspond to the linking constraint rows.



3. The initial right-hand side values specified for the linking constraints in each subproblem must be chosen so that, correspondingly, each subproblem has a feasible and finite optimal LP solution.
4. The number of rows (exclusive of objective and linking constraint rows) in each subproblem must be .LE. 10; the number of decision variables must be .LE. 20 in each subproblem.
5. The number of nonzero objective and constraint coefficients must be .LE. 500 in each subproblem; the total number in all subproblems must be .LE. 1000.
6. The number of distinct sets of optimal dual multipliers generated in each subproblem must be .LE. 20.
7. The number of distinct allocations explicitly examined in each subproblem must be .LE. 40.
8. The list of stored nodes in the branch-and-bound search for an optimal allocation of the linking resources must never exceed 200 nodes in length.
9. The list of stored nodes in the branch-and-bound search for optimal subproblem solutions must never exceed 500 nodes in length for any subproblem.

#### 4.3. Description of Subprograms

- 1-21. BLOCK DATA and SUBROUTINES INPUT(IFPROB), FTRAN(IPAR), BTRAN, FORMC, PRICE, CHUZR, WRETA, SHIFTR(IOLD, INEW), INVERT, UNPACK(IV), SHFTE, UPBETA, NORMAL, BANDB(INITBD), DCHUZR, DCHUZR, TESTX, PENLTS, BRANCH, BKTRAK--Same as for DSLC; see

Section 3.3. SUBROUTINE INPUT(IFPROB) modified for multiple linking constraints.

22. SUBROUTINE ALOCTE(IPAR)--Obtains linear programming solution to Relaxed Master Problem; see Kochman (1976), Section 5.1. Calls SUBROUTINE LDMSTR (if necessary) to set up problem data for solution by SUBROUTINE NORMAL. Also calls SUBROUTINE TESTZ to round-off optimal LP solution.

Parameters:

IPAR (Input, INTEGER\*4) Parameter indicating whether (IPAR = 0) or not (IPAR = 1) any subproblem has been explicitly solved since last Master Problem solution. When IPAR = 0, the data for the Master Problem must be reloaded from scratch (SUBROUTINE LDMSTR) into the storage locations required for SUBROUTINE NORMAL. When IPAR = 1, only the revised allocation-variables bounds need be reloaded.

23. SUBROUTINE LDMSTR--Loads data for Relaxed Master Problem into appropriate storage locations for solution by SUBROUTINE NORMAL.
24. SUBROUTINE TESTZ--Tests LP-optimal solution to Relaxed Master Problem for fathoming. If not fathomed, the solution is rounded to obtain an integer-feasible solution to the Relaxed Master Problem. See Kochman (1976), Section 5.1 for a discussion of the rounding procedure.
- 25-27. SUBROUTINES DOMCHK(J, K), LDDATA(J), GETLBD(J),--Same as for DSIC (see Section 3.3). Modified for multiple linking constraints.



28. SUBROUTINE PISAVR(J)--Tests current LP-optimal solution to subproblem J to determine if the set of optimal dual multipliers associated with the linking constraints is distinct from previous sets; stores these multipliers if distinct.

Parameters:

J (Input, INTEGER\*4) Indexes subproblem solved under current allocation.

29-32. SUBROUTINES SAVER(J), DMBRAN(J, K), BBKTRK, OUTSOL(INC)--Same as for DSLC (see Section 3.3). Modified for multiple linking constraints.

#### 4.4. Description of Output

The format for output from DMIC is identical to that for DSLC, with the obvious exception that no parametric linear programming solution information for the subproblems is generated in DMIC. The output from a sample run of DMIC is included in Appendix III, following the listing of the program.

#### 4.5. Further Comments

As currently written, the Fortran code DMIC contains approximately 2,000 source statements. All significant array storage is done in COMMON. In all, approximately 50K bytes of core storage are required by the five COMMON blocks.

The computer program DMIC was written by the present author in the course of completing his doctoral dissertation in the Department of Operations Research, Stanford University. DMIC was last revised in September 1976.

#### REFERENCES

- Beale, E.M.L., 1968, Mathematical Programming in Practice, Wiley and Sons, Inc., New York.
- Beale, E.M.L., and R.E. Small, 1965, "Mixed Integer Programming by a Branch-and-Bound Technique," Proceedings of the Third IFIP Congress, Vol. 2, pp. 450-451.
- Dakin, R.J., 1965, "A Tree Search Algorithm for Mixed Integer Programming Problems," Computer Journal, Vol. 8, pp. 250-255.
- Kochman, G.A., 1976, "Decomposition in Integer Programming," Technical Report, Department of Operations Research, Stanford University.
- Little, J.D.C., K.C. Murty, B.W. Sweeney, and C. Karel, 1963, "An Algorithm for the Travelling Salesman Problem," Operations Research, Vol. 11, pp. 972-989.
- Tomlin, J.A., 1970, "Branch-and-Bound Methods for Integer and Non-Convex Programming," in Integer and Nonlinear Programming, J. Abadie, ed., North Holland, Amsterdam.
- Tomlin, J.A., 1975, "Users Guide for LPM-1," Systems Optimization Laboratory, Department of Operations Research, Stanford University.



APPENDIX I

LISTING AND SAMPLE INPUT/OUTPUT FOR COMPUTER PROGRAM BB

```

1. C
2. C      BRANCH-AND-BOUND ROUTINE, BB
3. C      PURE-INTEGER LINEAR PROGRAMMING IN GENERAL INTEGER VARIABLES
4. C      GARY A. KUCHMAN (OPERATIONS RESEARCH), SEPT. 1976, MOD 1
5. C
6. C      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
7. C      1      INTEGER*4 (I-N,Q)
8. C      COMMON/TIMERS/ TFIRST,TOPT,IFVAL,ITOT
9. C
10. C      MAIN PROGRAM
11. C      COORDINATES APPLICATION OF INPUT, LP, BRANCH-AND-BOUND,
12. C      AND OUTPUT SUBROUTINES
13. C
14. C      START TIMER
15. C      10 CALL PCLOCK(ITOT)
16. C      INPUT PROBLEM DATA
17. C      CALL INPUT(IFPROB,INITBD)
18. C      IF (IFPROB .EQ. 0) GO TO 1000
19. C      ITSINV = 99999
20. C      SOLVE LP RELAXATION OF ORIGINAL PROBLEM
21. C      CALL ACRMAL(ITSINV)
22. C      APPLY BRANCH-AND-BOUND SEARCH ROUTINE
23. C      CALL BANDB(INITBD,ITSINV)
24. C      STOP TIMER
25. C      CALL PCLOCK(JTIME,ITOT)
26. C      TTOT = JTIME/100.
27. C      WRITE (6,1) TTOT
28. C      1 FORMAT (' TOTAL SOLUTION TIME =',F7.2,' SECONDS')
29. C      OUTPUT OPTIMAL SOLUTION
30. C      CALL WRAPUP
31. C      GO TO 10
32. C      1000 STOP
33. C      END
34. C      BLOCK DATA
35. C
36. C      INITIALIZES GLOBAL PROGRAM CONSTANTS
37. C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
38. C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
39. C
40. C      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
41. C      1      INTEGER*4 (I-N,Q)
42. C      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRC,QMA,QBA,QFI,
43. C      1      CEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
44. C      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
45. C      REAL ZTOLZE/1.E-5/,ZTOLPV/1.E-4/,ZTCOST/1.E-3/,ZTOLRJ/1.E-4/,
46. C      1      ZTOLSM/1.E-10/
47. C      INTEGER*4 NRMAX/60/,NTMAX/1000/,NEMAX/4000/
48. C      INTEGER*4 QRC/'RC',QA/'QA',QMA/'MATR',QBA/'BASI',QFI/'FIRS',
49. C      1      IQEO/'ECF',QBL/'',QA/'A',QPL/' + ',QMI/' - ',
50. C      2      QZ/' Z ',QI/' I ',QF/' F ',QN/' N ',QU/' U ',
51. C      3      QB/' B ',QC/' C ',QE/' E ',QH/' H ',QL/' L ',
52. C      4      QO/' O ',QR/' R ',QM/' M ',QG/' G '
53. C      END
54. C      SUBROUTINE INPUT(IFPROB,INITBD)
55. C
56. C      INPUTS PROBLEM DATA
57. C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
58. C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
59. C      *****DESCRIPTION OF PARAMETERS*****
60. C      IFPROB = INPROB = NONZERO PROBLEM IDENTIFICATION NUMBER(OUTPUT)

```



```

61. C          INITBD = INITIAL LOWER BOUND ESTIMATE FOR MAXIMAL OBJECTIVE
62. C          VALUE (OUTPUT)
63. C*****
64. C
65. C          IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
66. C          1          INTEGER*4 (I-N,Q)
67. C          INTEGER*2 JH,KINBAS,LA,LE,IA,IE
68. C          DOUBLE PRECISION E(4000),ATEMP1,ATEMP2
69. C          REAL A(2000)
70. C
71. C          COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRD,QMA,QBA,QFI,
72. C          1          QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
73. C          2          QE,QH,QL,QD,QR,QM,QG,NTMAX
74. C
75. C          COMMON DSUM,DPROD,CY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
76. C          1          SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,IOBJ,IROWP,
77. C          2          IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
78. C          3          NETA,NLELEM,NLETA,NUELEM,NUETA,JH(60),KINBAS(122),LA(122),
79. C          4          LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
80. C
81. C          DO 10 I=1,60
82. C          10 B(I) = 0.
83. C          DO 20 J=1,122
84. C          20 KINBAS(J) = 0
85. C          NROW = 0
86. C          ITCNT = 0
87. C          ICS1=C
88. C          ICS2=C
89. C          READ (5,7000,END=9999) INPROB,IOBJ,INVRQ,ITRFRQ,INITBD
90. C          7000 FORMAT (4I4,I10)
91. C          IFPROB = INPROB
92. C          IF (IFPROB .EQ. 0) RETURN
93. C          IF (ICBJ .EQ. 0) IOBJ = 1
94. C          IF (INVRQ .EQ. 0) INVRQ = 99999
95. C          IF (ITRFRQ .EQ. 0) ITRFRQ = 99999
96. C          WRITE(6,8010) IFPROB
97. C          8010 FORMAT(9H1PROBLEM ,I4)
98. C
99. C          5 READ(5,101) K1,K2,K3,K4,(NAME(I),I=1,4),ATEMP1,NAME(5),NAME(6),
100. C          1ATEMP2
101. C          101 FORMAT(4A1,2A4,2X,2A4,2X,F12.4,3X,2A4,2X,F12.4)
102. C          IF(K1 .EQ. QE) GO TO 600
103. C          IF(K1 .EQ. QBL) GO TO 50
104. C          IF(K1 .EQ. QN) GO TO 100
105. C          IF(K1 .EQ. QR .AND. K2 .EQ. QO) L=1
106. C          IF(K1 .EQ. QR .AND. K2 .EQ. QO) GO TO 150
107. C          IF(K1 .EQ. QC) L=2
108. C          IF(K1 .EQ. QC) GO TO 150
109. C          IF(K1 .EQ. QB .AND. K2 .EQ. QA) L=3
110. C          IF(K1 .EQ. QB .AND. K2 .EQ. QA) GO TO 150
111. C          IF(K1 .EQ. QF .AND. K2 .EQ. QH) L=4
112. C          IF(K1 .EQ. QF .AND. K2 .EQ. QH) GO TO 150
113. C          50 GO TO(210,320,410,500),L
114. C          100 NTEMP(1) = NAME(3)
115. C          NTEMP(2) = NAME(4)
116. C          150 CONTINUE
117. C          GO TO 5
118. C
119. C          210 NROW=NROW+1
120. C          NCOL=NROW
121. C          ICNAM(NROW,1) = NAME(1)

```

```

122.          ICNAM(NROW,2) = NAME(2)
123.      C
124.      C          TEST ROW TYPE
125.      C
126.          IF(K2.EQ.QL .OR. K3.EQ.QL) GO TO 220
127.          IF(K2.EQ.QE .OR. K3.EQ.QE) GO TO 230
128.          IF(K2.EQ.QG .OR. K3.EQ.QG) GO TO 240
129.          IF(K2.EQ.QN .OR. K3.EQ.QN) GO TO 250
130.          GO TO 230
131.      220 XLB(NROW) = 0.
132.          XUB(NROW) = 1.E4
133.          GO TO 250
134.      230 XLB(NROW) = 0.
135.          XUB(NROW) = 0.
136.          GO TO 250
137.      240 XLB(NROW) = 0.
138.          XUB(NROW) = 1.E4
139.          A(NROW) = -1.
140.          GO TO 260
141.      250 A(NROW) = 1.
142.      260 IA(NROW) = NROW
143.          LA(NROW) = NROW
144.          JH(NROW) = NROW
145.          KINBAS(NROW) = NROW
146.          NELEM=NROW
147.          GO TO 5
148.      C
149.      C          MATRIX ELEMENTS
150.      C
151.      320 J = 3
152.          K = 4
153.          IF (DABS(ATEMP1) .LE. ZTOLZE) GO TO 321
154.          GO TO 324
155.      321 J=5
156.          K=6
157.          ATEMP1=ATEMP2
158.          GO TO 330
159.      324 CONTINUE
160.          IF (NAME(1) .EQ. ICS1 .AND. NAME(2) .EQ. ICS2) GO TO 330
161.      C
162.      C          TEST FOR SPLIT VECTOR
163.      C
164.          DO 325 I = 1,NCOL
165.          IF (NAME(1) .NE. ICNAM(I,1)) GO TO 325
166.          IF (NAME(2) .NE. ICNAM(I,2)) GO TO 325
167.          WRITE(6,8250) NAME(1),NAME(2)
168.      8250 FORMAT(14HOSPLIT VECTOR ,2A4)
169.      325 CONTINUE
170.          NCOL = NCOL + 1
171.          ICS1 = NAME(1)
172.          ICS2 = NAME(2)
173.          ICNAM(NCOL,1) = ICS1
174.          ICNAM(NCOL,2) = ICS2
175.          LA(NCOL) = NELEM + 1
176.      C
177.      C          TEST FOR ROW MATCH
178.      C
179.      330 DO 340 I = 1,NROW
180.          IF(NAME(I) .NE. ICNAM(I,1) .OR. NAME(I) .NE. ICNAM(I,2))GO TO 340
181.          NELEM = NELEM + 1
182.          IA(NELEM) = I

```



```

183.      A(NELEM) = ATEMP1
184.      LA(NCCL+1)=NELEM+1
185. 335 IF(K .GT. 5) GO TO 5
186.      IF(DARS(ATEMP2) .LE. ZTOLZE) GO TO 5
187.      J = 5
188.      K = 6
189.      ATEMP1 = ATEMP2
190.      GO TO 330
191. 340 CONTINUE
192.      WRITE(6,8300) NAME(J),NAME(K),NAME(1),NAME(2)
193. 8300 FORMAT(18HNO MATCH FOR ROW ,2A4,10HAT COLUMN ,2A4)
194.      STOP
195. C
196. C      BASIS CARDS
197. C
198. 410 DO 420 I = 1,NCOL
199.      IF(NAME(I) .NE. ICNAM(I,1) .OR. NAME(2) .NE. ICNAM(I,2)) GO TO 420
200.      IBVEC = I
201.      GO TO 425
202. 420 CONTINUE
203.      WRITE(6,8400) NAME(1),NAME(2)
204. 8400 FORMAT(21HNO MATCH FOR VECTOR ,2A4)
205.      GO TO 5
206. 425 DO 430 I = 1,NROW
207.      IF (NAME(3) .NE. ICNAM(I,1) .OR. NAME(4) .NE. ICNAM(I,2)) GO TO 430
208.      IBROW=I
209.      GO TO 440
210. 430 CONTINUE
211.      WRITE(6,8300) NAME(3),NAME(4)
212.      GO TO 5
213. 440 JH(IBROW) = IBVEC
214.      KINBAS(IBROW) = 0
215.      KINBAS(IBVEC) = IBROW
216.      GO TO 5
217. C
218. C      RHS
219. C
220. 500 J = 3
221.      K = 4
222.      IF (DABS(ATEMP1) .LE. ZTOLZE) GO TO 521
223.      GO TO 524
224. 521 J=5
225.      K=6
226.      ATEMP1=ATEMP2
227.      GO TO 530
228. 524 CONTINUE
229.      IF (NAME(1) .EQ. ICS1 .AND. NAME(2) .EQ. ICS2) GO TO 530
230. C
231. C      TEST FOR SPLIT VECTOR
232. C
233. DO 525 I = 1,NCOL
234.     IF (NAME(I) .NE. ICNAM(I,1)) GO TO 525
235.     IF (NAME(2) .NE. ICNAM(I,2)) GO TO 525
236.     WRITE(6,8250) NAME(1),NAME(2)
237. 525 CONTINUE
238.     ICS1 = NAME(1)
239.     ICS2 = NAME(2)
240. C
241. C      TEST FOR ROW MATCH
242. C
243. 530 DO 540 I = 1,NROW

```

```

244.      IF(NAME(J) .NE. ICNAM(1,1) .OR. NAME(K) .NE. ICNAM(1,2))GO TO 540
245.      B(I) = ATEMP1
246.      535 IF(K .GT. 5) GO TO 5
247.      IF(DABS(ATEMP2) .LE. ZTOLZE) GO TO 5
248.      J = 5
249.      K = 6
250.      ATEMP1 = ATEMP2
251.      GO TO 530
252.      540 CONTINUE
253.      WRITE(6,8300) NAME(J),NAME(K)
254.      STOP
255.      C
256.      C          END OF INPUT
257.      C
258.      600 NSCOL = NCOL - NKOW
259.      K = NROW + 1
260.      C          INPUT LOWER AND UPPER BOUNDS ON DECISION VARIABLES
261.      READ (5,650) (XLB(J), J=K,NCOL)
262.      READ (5,650) (XUB(J), J=K,NCOL)
263.      650 FORMAT (15F5.0)
264.      NELEM = NELEM - NKOW
265.      RELEM = NELEM
266.      RDENS = RELEM/(NROW*NSCOL)
267.      RETURN
268.      9999 IFPROB = 0
269.      RETURN
270.      END
271.      SUBROUTINE FTRAN(IPAR)
272.      C
273.      C          PERFORMS FORWARD TRANSFORMATION ON COLUMN STORED IN VECTOR Y
274.      C          SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
275.      C          BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
276.      C*****DESCRIPTION OF PARAMETERS*****
277.      C          IPAR = PARAMETER INDICATING WHICH ETA-VECTORS MATRIX COLUMN
278.      C          IS TO BE UPDATED BY (INPUT)
279.      C*****
280.      C
281.      C          IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
282.      C          INTEGER*4 (I-N,Q)
283.      C          INTEGER*2 JH,KINBAS,LA,LE,IA,IE
284.      C          DOUBLE PRECISION E(4000)
285.      C          REAL A(2000)
286.      C
287.      C          COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
288.      C          1          QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
289.      C          2          QE,QH,QL,QO,QR,QM,QG,NTMAX
290.      C
291.      C          COMMON DSUM,DPRGD,DY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
292.      C          1          SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,IOBJ,IROWP,
293.      C          2          IVIN,IROUT,ITCNT,INVRQ,ITRFRQ,IFFEZ,JCOLP,NKOW,NCOL,NELEM,
294.      C          3          NETA,NLELEM,NLETA,NUELEM,NUETA,JH(60),KINBAS(122),LA(122),
295.      C          4          LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
296.      C
297.      GO TO (100,110), IPAR
298.      100 NFE = 1
299.      NLE = NETA
300.      GO TO 200
301.      110 NFE = NLETA + 1
302.      NLE = NETA
303.      200 IF (NFE .GT. NLE) GO TO 9000
304.      DO 1000 IK = NFE,NLE

```



```

305.      LL = LE(IK)
306.      KK = LE(IK+1) - 1
307.      IPIV = IE(LL)
308.      DY = Y(IPIV)
309.      DY = DY/E(LL)
310.      Y(IPIV) = DY
311.      IF (KK .LE. LL) GO TO 1000
312.      LL = LL + 1
313.      DO 500 J = LL, KK
314.      IR = IE(J)
315.      Y(IR) = Y(IR) - E(J) * DY
316.      500 CONTINUE
317.      1000 CONTINUE
318.      9000 CONTINUE
319.      RETURN
320.      END
321.      SUBROUTINE BTRAN
322.      C
323.      C      PERFORMS BACKWARD TRANSFORMATION ON COLUMN STORED IN VECTOR Y
324.      C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
325.      C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
326.      C
327.      C      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
328.      1      INTEGER*4 (I-N,Q)
329.      C      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
330.      C      DOUBLE PRECISION E(4000)
331.      C      REAL A(2000)
332.      C
333.      C      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
334.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
335.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
336.      C
337.      C      COMMON DSUM,DPROD,DY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
338.      1      SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,I0BJ,IROWP,
339.      2      IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
340.      3      NETA,NLELEM,NLETA,NUELEM,NUETA,JH(60),KINBAS(122),LA(122),
341.      4      LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
342.      C
343.      IF (NETA .LE. 0) GO TO 9000
344.      DO 1000 I = 1, NETA
345.      IK = NETA - I + 1
346.      LL = LE(IK)
347.      KK = LE(IK+1) - 1
348.      IPIV = IE(LL)
349.      DP = E(LL)
350.      DY = Y(IPIV)
351.      DSUM = 0.
352.      IF (KK .LE. LL) GO TO 600
353.      LL = LL + 1
354.      DO 500 J = LL, KK
355.      IR = IE(J)
356.      DE = E(J)
357.      DPROD = DE * Y(IR)
358.      DSUM = DSUM + DPROD
359.      500 CONTINUE
360.      C
361.      600 Y(IPIV) = (DY - DSUM) / DP
362.      1000 CONTINUE
363.      9000 RETURN
364.      END
365.      SUBROUTINE FORMC

```

```

366. C
367. C      FORMS OBJ. FUNCTION VECTOR; USED IN SUBROUTINE PRICE
368. C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
369. C      BY J.A. TUMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
370. C
371.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
372.      1      INTEGER*4 (I-N,Q)
373.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
374.      DOUBLE PRECISION E(4000)
375.      REAL A(2000)
376. C
377.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCCST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
378.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
379.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
380. C
381.      COMMON DSUM,DPROD,CY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
382.      1      SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,IOBJ,IROWP,
383.      2      IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
384.      3      NETA,NLELEM,NLETA,NUELEM,NUETA,JH(60),KINBAS(122),LA(122),
385.      4      LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
386. C
387.      MSTAT = QI
388.      IFFEZ = 0
389.      SUMINF = 0.
390.      Y(IOBJ) = 0.
391.      DO 30 I=1,NROW
392.      IF (I.EQ. IOBJ) GO TO 30
393.      ICOL = JH(I)
394.      IF (X(I) .LE. (XLB(ICOL) - ZTOLZE)) GO TO 10
395.      IF (X(I) .GE. (XUB(ICOL) + ZTOLZE)) GO TO 20
396.      Y(I) = 0.
397.      GO TO 30
398. 10 Y(I) = 1.
399.      SUMINF = SUMINF + XLB(ICOL) - X(I)
400.      GO TO 30
401. 20 Y(I) = -1.
402.      SUMINF = SUMINF + X(I) - XUB(ICOL)
403. 30 CONTINUE
404.      IF (SUMINF .GT. ZTOLSM) RETURN
405.      Y(IOBJ) = 1.
406.      IFFEZ = 1
407.      MSTAT = QF
408.      RETURN
409.      END
410.      SUBROUTINE PRICE
411. C
412. C      PRICES OUT NONBASIC COLUMNS; CHOOSES PIVOT COLUMN JCOLP FOR
413. C      CURRENT PRIMAL SIMPLEX ITERATION
414. C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
415. C      BY J.A. TUMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
416. C
417.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
418.      1      INTEGER*4 (I-N,Q)
419.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
420.      INTEGER*2 IPART,INCUMB,IVBND,IVID,IOBND
421.      DOUBLE PRECISION E(4000)
422.      REAL A(2000)
423. C
424.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCCST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
425.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
426.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX

```



```

427. C
428. COMMON/BLOCK2/ DEPART(60),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
429. 1 IPART(122),INCUMB(122),IVBND(500),IVID(500),
430. 2 IUBND(500),NP1VOT,IPTYPE,RCOST,IFEAS
431. C
432. COMMON DSUM,DPROD,OY,DE,OP,B(60),X(60),Y(60),YTEMP(60),A,E,
433. 1 SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,IOBJ,IROWP,
434. 2 IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
435. 3 NETA,NLELEM,NLETA,NUELEM,NUETA,JH(60),KINBAS(122),LA(122),
436. 4 LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
437. C
438. CMIN = 1.E10
439. CMAX = -1.E10
440. DO 1000 J=1,NCOL
441. IF (KINBAS(J) .GT. 0) GO TO 1000
442. IF ((XUB(J) - XLB(J)) .LT. 2TOLZE) GO TO 1000
443. DSUM = 0.
444. LL = LA(J)
445. KK = LA(J+1) - 1
446. DO 500 I = LL, KK
447. IR = IA(I)
448. DE = A(I)
449. DPROD = DE * Y(IR)
450. DSUM = DSUM + DPROD
451. 500 CONTINUE
452. IF (KINBAS(J) .EQ. -1) GO TO 600
453. IF (DSUM .GE. CMIN) GO TO 1000
454. CMIN = DSUM
455. JCOL1 = J
456. GO TO 1000
457. 600 IF (DSUM .LE. CMAX) GO TO 1000
458. CMAX = DSUM
459. JCOL2 = J
460. 1000 CONTINUE
461. C
462. IF (CMIN .LE. -2TCOST) GO TO 1500
463. IF (CMAX .GE. 2TCOST) GO TO 2000
464. JCOLP = 0
465. RCOST = 0.
466. RETURN
467. 1500 IF (CMAX .GE. 2TCOST) GO TO 2500
468. 1600 JCOLP = JCOL1
469. RCOST = CMIN
470. RETURN
471. 2000 JCOLP = JCOL2
472. RCOST = CMAX
473. RETURN
474. 2500 IF (ABS(CMIN) - CMAX) 2000,2000,1600
475. END
476. SUBROUTINE CHUZR
477. C
478. C PERFORMS MIN-RATIO TEST FOR PIVOT COLUMN JCOLP DETERMINED IN
479. C SUBROUTINE PRICE. SELECTS PIVOT ROW IROWP FOR CURRENT PRIMAL
480. C SIMPLEX ITERATION.
481. C SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
482. C BY J.A. TUMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
483. C
484. IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
485. 1 INTEGER*4 (I-N,Q)
486. INTEGER*2 JH,KINBAS,LA,LE,IA,IE
487. INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND

```

```

488.      DOUBLE PRECISION E(4000)
489.      REAL A(2000)
490.      C
491.      COMMON/BLUCK/ ZTOLZE,ZTOLPV,ZTCGST,ZTOLRJ,ZTOLSM,QRC,QMA,QBA,QFI,
492.      1 QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
493.      2 QE,QH,QL,QO,QR,QM,QG,NTMAX
494.      C
495.      COMMON/BLUCKZ/ DEPART(60),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
496.      1 IPART(122),INCUMB(122),IVBND(500),IVID(500),
497.      2 IUBND(500),NPIVOT,IPTYPE,RCGST,IFEAS
498.      C
499.      COMMON DSUM,DPROD,DY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
500.      1 SUMINF,IGNAM(122,2),NAME(20),NTEMP(20),MSAT,IOBJ,IROWP,
501.      2 IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
502.      3 NETA,NLELEM,NLETA,NUELEM,NUEA,JH(60),KINBAS(122),LA(122),
503.      4 LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
504.      C
505.      IF (KINBAS(JCOLP) .EQ. -1) GO TO 1000
506.      C
507.      C      INCCMG VARIABLE AT LOWER BOUND
508.      C
509.      DP = 1.E10
510.      DO 500 I=1,NROW
511.      IF (I .EQ. IOBJ) GO TO 500
512.      ICOL = JH(I)
513.      IF (Y(I) .GT. ZTOLPV) GO TO 100
514.      IF (Y(I) .LT. -ZTOLPV) GO TO 200
515.      GO TO 500
516.      C      POSITIVE COEFFICIENT
517.      100 IF (X(I) .LT. (XLB(ICOL) - ZTOLZE)) GO TO 500
518.      DE = (X(I) - XLB(ICOL))/Y(I)
519.      IF (DE .GE. DP) GO TO 500
520.      IPTYPE = 0
521.      GO TO 250
522.      C      NEGATIVE COEFFICIENT
523.      200 IF (X(I) .GT. (XUB(ICOL) + ZTOLZE)) GO TO 500
524.      DE = (X(I) - XUB(ICOL))/Y(I)
525.      IF (DE .GE. DP) GO TO 500
526.      IPTYPE = -1
527.      250 DP = DE
528.      IROWP = I
529.      500 CONTINUE
530.      DE = DP + XLB(JCOLP)
531.      IF (DE .LT. XUB(JCOLP)) GO TO 600
532.      DP = XUB(JCOLP) - XLB(JCOLP)
533.      NPIVOT = 0
534.      RETURN
535.      600 NPIVOT = 1
536.      RETURN
537.      C
538.      C      INCCMG VARIABLE AT UPPER BOUND
539.      C
540.      1000 DP = -1.E10
541.      DO 1500 I=1,NROW
542.      IF (I .EQ. IOBJ) GO TO 1500
543.      ICOL = JH(I)
544.      IF (Y(I) .GT. ZTOLPV) GO TO 1100
545.      IF (Y(I) .LT. -ZTOLPV) GO TO 1200
546.      GO TO 1500
547.      C      POSITIVE COEFFICIENT
548.      1100 IF (X(I) .GT. (XUB(ICOL) + ZTOLZE)) GO TO 1500

```



```

549.      DE = (X(I) - XUB(ICOL))/Y(I)
550.      IF (DE .LE. DP) GO TO 1500
551.      IPTYPE = -1
552.      GO TO 1250
553.  C      NEGATIVE COEFFICIENT
554.      1200 IF (X(I) .LT. (XLB(ICOL) - ZTOLZE)) GO TO 1500
555.      DE = (X(I) - XLB(ICOL))/Y(I)
556.      IF (DE .LE. DP) GO TO 1500
557.      IPTYPE = 0
558.      1250 DP = DE
559.      IROWP = I
560.      1500 CONTINUE
561.      DE = DP + XUB(JCOLP)
562.      IF (DE .GT. XLB(JCOLP)) GO TO 1600
563.      DP = XLB(JCOLP) - XUB(JCOLP)
564.      NPIVOT = 0
565.      RETURN
566.      1600 NPIVOT = 1
567.      RETURN
568.      END
569.      SUBROUTINE WRETA
570.  C
571.  C      FORMS NEW ETA-VECTORS FOR PRODUCT FORM OF BASIS INVERSE
572.  C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
573.  C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
574.  C
575.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
576.      1      INTEGER*4 (I-N,Q)
577.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
578.      DOUBLE PRECISION E(4000)
579.      REAL A(2000)
580.  C
581.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
582.      1      QED,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
583.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
584.  C
585.      COMMON DSUM,DPROD,DY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
586.      1      SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,IOBJ,IROWP,
587.      2      IVIN,IVOUT,ITCNT,INVERQ,ITFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
588.      3      NETA,NLELEM,NLETA,NUELEM,NUETA,H(60),KINBAS(122),LA(122),
589.      4      LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
590.  C
591.      NELEM = NELEM + 1
592.      IE(NELEM) = IROWP
593.      E(NELEM) = Y(IROWP)
594.  C
595.      DO 1000 I = 1,NROW
596.      IF (I .EQ. IROWP) GO TO 1000
597.      IF (DABS(Y(I)) .LE. ZTOLZE) GO TO 1000
598.      NELEM = NELEM + 1
599.      IE(NELEM) = I
600.      E(NELEM) = Y(I)
601.      1000 CONTINUE
602.  C
603.      NETA = NETA + 1
604.      LE(NELEM+1) = NELEM + 1
605.      RETURN
606.      END
607.      SUBROUTINE SHIFTR(IOLD,INEW)
608.  C
609.  C      REARRANGES DATA STORAGE

```

```

610. C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
611. C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
612. C*****DESCRIPTION OF PARAMETERS*****
613. C      IOLD, INEW = PARAMETERS INDEXING STORAGE LOCATIONS IN WHICH
614. C      DATA IS TO BE TRANSFERRED (INPUT)
615. C*****
616. C
617. C      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
618. C      1      INTEGER*4 (I-N,Q)
619. C      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
620. C      DOUBLE PRECISION E(4000)
621. C      REAL A(2000)
622. C
623. C      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
624. C      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
625. C      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
626. C
627. C      COMMON DSUM,DPROD,DY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
628. C      1      SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,IOBJ,IRUWP,
629. C      2      IVIN,IVOUT,ITCNT,INVERQ,ITFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
630. C      3      NETA,NLELEM,NLETA,NUELEM,NUETA,JH(60),KINBAS(122),LA(122),
631. C      4      LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
632. C
633. C      DIMENSION BARRAY(240)
634. C      EQUIVALENCE (BARRAY(1),B(1))
635. C      IFO = (IOLD - 1) * NRMAX
636. C      IFN = (INEW - 1) * NRMAX
637. C
638. C      DO 1000 I = 1,NROW
639. C      BARRAY(IFO + I) = BARRAY(IFN + I)
640. C      1000 CONTINUE
641. C      RETURN
642. C      END
643. C      SUBROUTINE INVERT
644. C
645. C      COMPUTES INVERSE OF CURRENT BASIS BY LU DECOMPOSITION
646. C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
647. C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
648. C
649. C      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
650. C      1      INTEGER*4 (I-N,Q)
651. C      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
652. C      DOUBLE PRECISION E(4000)
653. C      REAL A(2000)
654. C
655. C      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
656. C      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
657. C      2      QE,QH,QL,QU,QR,QM,QG,NTMAX
658. C
659. C      COMMON DSUM,DPROD,DY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
660. C      1      SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,IOBJ,IRUWP,
661. C      2      IVIN,IVOUT,ITCNT,INVERQ,ITFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
662. C      3      NETA,NLELEM,NLETA,NUELEM,NUETA,JH(60),KINBAS(122),LA(122),
663. C      4      LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
664. C
665. C      INTEGER*2 MREG,HREG,VREG
666. C      DIMENSION MREG(60),HREG(60),VREG(60)
667. C      EQUIVALENCE (MREG(1),YTEMP(1)),(HREG(1),YTEMP(31)),(VREG(1),X(1))
668. C
669. C      SET PARAMETERS
670. C

```



```

671.      NETA = 0
672.      NLETA = 0
673.      NUETA = 0
674.      NELEM = 0
675.      NLELEM = 0
676.      NUELEM = 0
677.      NABOVE = 0
678.      LE(1) = 1
679.      LR1 = 1
680.      KR1 = 0
681.      LR4 = NROW + 1
682.      KR4 = NROW
683.
684.      C      PUT SLACKS AND ARTIFICIALS IN PART 4 AND REST IN PART 1
685.      C
686.      DO 100 I = 1,NROW
687.      IF (JH(I) .GT. NROW) GO TO 50
688.      LR4 = LR4 - 1
689.      VREG(LR4) = JH(I)
690.      VREG(LR4) = JH(I)
691.      GO TO 90
692.      50 KR1 = KR1 + 1
693.      VREG(KR1) = JH(I)
694.      50 HREG(I) = -1
695.      JH(I) = 0
696.      100 CONTINUE
697.      C
698.      KR3 = LR4 - 1
699.      LR3 = LR4
700.      C
701.      DO 200 I = LR4,KR4
702.      IR = VREG(I)
703.      HREG(IR) = 0
704.      JH(IR) = IR
705.      KINBAS(IR) = IR
706.      200 CONTINUE
707.      C
708.      C      PULL OUT VECTORS BELOW BUMP AND GET ROW COUNTS
709.      C
710.      NBNONZ = KR4 - LR4 + 1
711.      IF (KR1 .EQ. 0) GO TO 1190
712.      J = LR1
713.      210 IV = VREG(J)
714.      LL = LA(IV)
715.      KK = LA(IV+1) - 1
716.      IRCNT = 0
717.      DO 220 I = LL, KK
718.      NBNONZ = NBNONZ + 1
719.      IR = IA(I)
720.      IF (HREG(IR) .GE. 0) GO TO 220
721.      IRCNT = IRCNT + 1
722.      HREG(IR) = HREG(IR) - 1
723.      IRP = IR
724.      220 CONTINUE
725.      IF (IRCNT - 1) 230,250,300
726.      230 WRITE(6,8000)
727.      8000 FORMAT(16HOMATRIX SINGULAR )
728.      KINBAS(IV) = 0
729.      VREG(J) = VREG(KR1)
730.      KR1 = KR1 - 1
731.      IF (J .GT. KR1) GO TO 310

```

```

732.      GO TO 210
733.      C
734.      250 VREG(J) = VREG(KR1)
735.      KR1 = KR1 - 1
736.      LR3 = LR3 - 1
737.      VREG(LR3) = IV
738.      MREG(LR3) = IRP
739.      HREG(IRP) = 0
740.      JH(IRP) = IV
741.      KINBAS(IV) = IRP
742.      IF (J .GT. KR1) GO TO 310
743.      GO TO 210
744.      300 IF (J .GE. KR1) GO TO 310
745.      J = J+1
746.      GO TO 210
747.      C
748.      C      PULL OUT REMAINING VECTORS ABOVE AND BELOW THE
749.      C      BUMP AND ESTABLISH MERIT COUNTS OF COLUMNS
750.      C
751.      310 NVREM = 0
752.      IF(KR1 .EQ. 0) GO TO 1190
753.      J = LR1
754.      320 IV = VREG(J)
755.      LL = LA(IV)
756.      KK = LA(IV+1) - 1
757.      IRCNT = 0
758.      DO 800 I = LL, KK
759.      IR = IA(I)
760.      IF(HREG(IR) .NE. -2) GO TO 400
761.      C
762.      C      PIVOT ABOVE BUMP (PART OF L)
763.      C
764.      NABOVE = NABOVE + 1
765.      IROWP = IR
766.      CALL UNPACK(IV)
767.      CALL WRETA
768.      NLETA = NETA
769.      JH(IR) = IV
770.      KINBAS(IV) = IR
771.      VREG(J) = VREG(KR1)
772.      KR1 = KR1 - 1
773.      NVREM = NVREM + 1
774.      HREG(IR) = IV
775.      GO TO 940
776.      C
777.      400 IF (HREG(IR) .GE. 0) GO TO 800
778.      IRCNT = IRCNT + 1
779.      IRP = IR
780.      800 CONTINUE
781.      C
782.      IF (IRCNT - 1) 810,900,1000
783.      810 WRITE(6,8000)
784.      KINBAS(IV) = 0
785.      VREG(J) = VREG(KR1)
786.      NVREM = NVREM + 1
787.      KR1 = KR1 - 1
788.      IF (J .GT. KR1) GO TO 1010
789.      GO TO 320
790.      C
791.      C      PUT VECTOR BELOW BUMP
792.      C

```



```

793.      900 VREG(J) = VREG(KR1)
794.      NVREM = NVREM + 1
795.      KR1 = KR1 - 1
796.      LR3 = LR3 - 1
797.      VREG(LR3) = IV
798.      MREG(LR3) = IRP
799.      HREG(IRP) = 0
800.      JH(IRP) = IV
801.      KINBAS(IV) = IRP
802.  C
803.  C      CHANGE ROW COUNTS
804.  C
805.      940 DO 950 II = LL, KK
806.          IIR = IA(II)
807.          IF (HREG(IIR) .GE. 0) GO TO 950
808.          HREG(IIR) = HREG(IIR) + 1
809.      950 CONTINUE
810.          IF (J .GT. KR1) GO TO 1010
811.          GO TO 320
812.      1000 IF (J .GE. KR1) GO TO 1010
813.          J = J+1
814.          GO TO 320
815.      1010 IF (NVREM .GT. 0) GO TO 310
816.  C
817.  C      GET MERIT COUNTS
818.  C
819.      1020 IF (KR1 .EQ. 0) GO TO 1190
820.          DO 1100 J = LR1, KR1
821.              IV = VREG(J)
822.              LL = LA(IV)
823.              KK = LA(IV+1) - 1
824.              IMCNT = 0
825.              DO 1050 I = LL, KK
826.                  IR = IA(I)
827.                  IF (HREG(IR) .GE. 0) GO TO 1050
828.                  IMCNT = IMCNT - (HREG(IR) + 1)
829.          1050 CONTINUE
830.          MREG(J) = IMCNT
831.      1100 CONTINUE
832.  C
833.  C      SORT COLUMNS INTO MERIT ORDER
834.  C      USING SHELL SORT
835.  C
836.          ISD = 1
837.      1106 IF (KR1 .LT. 2*ISD) GO TO 1108
838.          ISD = 2*ISD
839.          GO TO 1106
840.      1108 ISD = ISD - 1
841.  C      END OF INITIALIZATION
842.      1101 IF (ISD .LE. 0) GO TO 1107
843.          ISK = 1
844.      1102 ISJ = ISK
845.          ISL = ISK + ISD
846.          ISY = MREG(ISL)
847.          ISZ = VREG(ISL)
848.      1103 IF (ISY .LT. MREG(ISJ)) GO TO 1104
849.      1105 ISL = ISJ + ISD
850.          MREG(ISL) = ISY
851.          VREG(ISL) = ISZ
852.          ISK = ISK + 1
853.          IF ((ISK + ISD) .LE. KR1) GO TO 1102

```

```

854.      ISD = (ISD - 1) / 2
855.      GO TO 1101
856.      1104 ISL = ISJ + ISD
857.      MREG(ISL) = MREG(ISJ)
858.      VREG(ISL) = VREG(ISJ)
859.      ISJ = ISJ - ISD
860.      IF (ISJ .GT. 0) GO TO 1103
861.      GO TO 1105
862.      1107 CONTINUE
863.      C      END OF SORT ROUTINE
864.      C
865.      C      PUT OUT BELOW BUMP ETAS (PART OF U)
866.      C
867.      1150 NSLCK = 0
868.      NBELOW = 0
869.      NELAST = NEMAX
870.      NTLAST = NTMAX
871.      LE(NTLAST + 1) = NELAST + 1
872.      C
873.      LR = LR3
874.      IF (LR3 .GE. LR4) LR = LR4
875.      IF (LR .GT. KR4) GO TO 2050
876.      JK = KR4 + 1
877.      DO 2000 JJ = LR, KR4
878.      JK = JK - 1
879.      IV = VREG(JK)
880.      I = MREG(JK)
881.      NBELOW = NBELOW + 1
882.      IF (IV .GT. NROW) GO TO 1200
883.      NSLCK = NSLCK + 1
884.      1200 LL = LA(IV)
885.      KK = LA(IV+1) - 1
886.      IF (KK .GT. LL) GO TO 1300
887.      1250 IF (ABS(A(LL) - 1.) .LE. ZTOLZE) GO TO 2000
888.      C
889.      1300 NUETA = NUETA + 1
890.      DO 1400 J = LL, KK
891.      IR = IA(J)
892.      IF (IR .EQ. I) GO TO 1390
893.      IE(NELAST) = IR
894.      E(NELAST) = A(J)
895.      NELAST = NELAST + 1
896.      NUELEM = NUELEM + 1
897.      GO TO 1400
898.      1390 EP = A(J)
899.      1400 CONTINUE
900.      IE(NELAST) = I
901.      E(NELAST) = EP
902.      LE(NTLAST) = NELAST
903.      NELAST = NELAST - 1
904.      NTLAST = NTLAST - 1
905.      NUELEM = NUELEM + 1
906.      2000 CONTINUE
907.      2050 IF (KRI .EQ. 0) GO TO 3500
908.      C
909.      C      DO L-U DECOMPOSITION OF BUMP
910.      C
911.      DO 3000 J = LRI, KRI
912.      IV = VREG(J)
913.      CALL UNPACK(IV)
914.      CALL FTRAN(2)

```

```

915.      IROWP = 0
916.      IRCMIN = -999999
917.      DO 2100 I = 1, NROW
918.      IF (DABS(Y(I)) .LE. ZTOLPV) GO TO 2100
919.      IF (HREG(I) .GE. 0) GO TO 2100
920.      IF (HREG(I) .LE. IRCMIN) GO TO 2100
921.      IRCMIN = HREG(I)
922.      IROWP = I
923. 2100 CONTINUE
924.      IF (IROWP .GT. 0) GO TO 2150
925.      WRITE(6,8000)
926.      KINBAS(IV) = 0
927.      GO TO 3000
928.  C
929. 2150 INCP = HREG(IROWP) + 3
930.  C
931.  C      WRITE L AND U ETAS
932.  C
933.      IF (J .EQ. KR1) GO TO 2160
934.      NELEM = NELEM + 1
935.      IE(NELEM) = IROWP
936.      E(NELEM) = Y(IROWP)
937. 2160 DO 2300 I = 1, NROW
938.      IF (I .EQ. IROWP) GO TO 2300
939.      IF (DABS(Y(I)) .LE. ZTOLZE) GO TO 2300
940.      IF (HREG(I) .GE. 0) GO TO 2200
941.  C
942.  C      L ETA ELEMENTS
943.  C
944.      NELEM = NELEM + 1
945.      IE(NELEM) = I
946.      E(NELEM) = Y(I)
947.      GO TO 2300
948.  C
949.  C      U ETA ELEMENTS
950.  C
951. 2200 IF(NELAST) = I
952.      E(NELAST) = Y(I)
953.      NELAST = NELAST + 1
954.      NUELEM = NUELEM + 1
955. 2300 CONTINUE
956.  C
957.      JH(IROWP) = IV
958.      KINBAS(IV) = IROWP
959.      NUETA = NUETA + 1
960.      IE(NELAST) = IROWP
961.      IF (J .NE. KR1) GO TO 2330
962.      E(NELAST) = Y(IROWP)
963.      GO TO 2340
964. 2330 E(NELAST) = 1.
965.      NETA = NETA + 1
966.      LE(NETA+1) = NELEM + 1
967. 2340 NUELEM = NUELEM + 1
968.      LE(NLAST) = NELAST
969.      NELAST = NELAST + 1
970.      NTLAST = NTLAST + 1
971.  C
972.  C      UPDATE ROW COUNTS
973.  C
974.      DO 2350 I = 1, NROW
975.      IF (DABS(Y(I)) .LE. ZTOLZE) GO TO 2350

```



```

976.      IF (HREG(I) .GE. 0) GO TO 2350
977.      HREG(I) = HREG(I) - INCR
978.      IF (HREG(I) .GE. 0) HREG(I) = -1
979.      2350 CONTINUE
980.      HREG(IPOWP) = 0
981.      3000 CONTINUE
982.      C
983.      C          MERGE L AND U ETAS
984.      C
985.      3500 NLETA = NETA
986.      NETA = NLETA + NUETA
987.      NLEEM = NELEM
988.      NELEM = NLEEM + NUELEM
989.      IF (NUELEM .EQ. 0) GO TO 3550
990.      CALL SHFTE
991.      C
992.      C          INSERT SLACKS FOR DELETED COLUMNS
993.      C
994.      3550 DO 3600 I = 1, NR0W
995.      IF (JH(I) .NE. 0) GO TO 3600
996.      JH(I) = 1
997.      IROWP = I
998.      CALL UNPACK(I)
999.      CALL FTRAN(1)
1000.     CALL WETA
1001.     3600 CONTINUE
1002.     C
1003.     C          UPDATE X
1004.     C
1005.     CALL SHIFTR(1,3)
1006.     DO 9000 J=1,NCOL
1007.     IF (KINBAS(J)) 8600,8700,9000
1008.     8600 DE = XUB(J)
1009.     GO TO 8750
1010.     8700 DE = XLB(J)
1011.     8750 LL = LA(J)
1012.     KK = LA(J+1) - 1
1013.     DO 8800 I=LL,KK
1014.     IR = IA(I)
1015.     8800 Y(IR) = Y(IR) - A(I)*DE
1016.     9000 CONTINUE
1017.     CALL FTRAN(1)
1018.     CALL SHIFTR(3,2)
1019.     C
1020.     C          PRINT STATISTICS
1021.     C
1022.     NOFD = NELEM - NETA
1023.     NSTR = NR0W - NSLCK
1024.     C
1025.     RETURN
1026.     END
1027.     SUBROUTINE UNPACK(IV)
1028.     C
1029.     C          EXPANDS COMPRESSED MATRIX COLUMNS
1030.     C          SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
1031.     C          BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1032.     C*****DESCRIPTION OF PARAMETERS*****
1033.     C          IV = PARAMETER INDEXING COLUMN TO BE EXPANDED (INPUT)
1034.     C*****
1035.     C
1036.     IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),

```

```

1037.      1      INTEGER*4 (I-N,Q)
1038.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1039.      DOUBLE PRECISION E(4000)
1040.      REAL A(2000)
1041.      C
1042.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1043.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1044.      2      QE,QH,QL,QQ,QK,QM,QG,NTMAX
1045.      C
1046.      COMMON DSUM,DPROD,DY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
1047.      1      SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,IOBJ,IRWP,
1048.      2      IVIN,IVOUT,ITCNT,INVERQ,ITFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
1049.      3      NETA,NLELEM,NLETA,NUELEM,NUETA,JH(60),KINBAS(122),LA(122),
1050.      4      LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
1051.      C
1052.      DO 100 I = 1,NROW
1053.      Y(I) = 0.
1054.      100 CONTINUE
1055.      C
1056.      LL = LA(IV)
1057.      KK = LA(IV+1) - 1
1058.      DO 200 I = LL, KK
1059.      IR = IA(I)
1060.      Y(IR) = A(I)
1061.      200 CONTINUE
1062.      C
1063.      RETURN
1064.      END
1065.      SUBROUTINE SHETE
1066.      C
1067.      C      SUBROUTINE FOR INVERT
1068.      C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
1069.      C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1070.      C
1071.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1072.      1      INTEGER*4 (I-N,Q)
1073.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1074.      DOUBLE PRECISION E(4000)
1075.      REAL A(2000)
1076.      C
1077.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1078.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1079.      2      QE,QH,QL,QQ,QR,QM,QG,NTMAX
1080.      C
1081.      COMMON DSUM,DPROD,DY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
1082.      1      SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,IOBJ,IRWP,
1083.      2      IVIN,IVOUT,ITCNT,INVERQ,ITFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
1084.      3      NETA,NLELEM,NLETA,NUELEM,NUETA,JH(60),KINBAS(122),LA(122),
1085.      4      LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
1086.      C
1087.      C      SHIFT IE AND E OF J ELEMENTS
1088.      C
1089.      NF = NEMAX - NUELEM + 1
1090.      INCR = 0
1091.      DO 1000 I = NF,NEMAX
1092.      INCR = INCR + 1
1093.      IE(NLELEM + INCR) = IE(I)
1094.      E(NLELEM + INCR) = E(I)
1095.      1000 CONTINUE
1096.      C
1097.      IDIF = NEMAX - NLELEM - NUELEM

```

```

1098.      NF = NMAX - NLETA + 1
1099.      INCR = 0
1100.      DO 2000 I = NF, NMAX
1101.      INCR = INCR + 1
1102.      LE(NLETA + INCR) = LE(I) - IDIF
1103. 2000 CONTINUE
1104.      LE(NLETA+1) = NELEM + 1
1105.      RETURN
1106.      END
1107.      SUBROUTINE UPBETA
1108.
1109.  C
1110.  C      UPDATES RIGHT-HAND SIDES TO REFLECT NEW BASIS RESULTING FROM
1111.  C      CURRENT SIMPLEX PIVOT
1112.  C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
1113.  C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1114.  C
1115.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1116. 1      INTEGER*4 (I-N,Q)
1117.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1118.      INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
1119.      DOUBLE PRECISION E(4000)
1120.      REAL A(2000)
1121.  C
1122.      COMMON/BLOCK1/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QF1,
1123. 1      QEO,QSL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1124. 2      QE,QH,QL,QQ,QR,QM,QG,NMAX
1125.  C
1126.      COMMON/BLOCK2/ DEPART(60),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1127. 1      IPART(122),INCUMB(122),IVBND(500),IVID(500),
1128. 2      ICBND(500),NPIVOT,IPTYPE,RCOST,IFEAS
1129.  C
1130.      COMMON DSUM,DPROD,DY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
1131. 1      SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MS,TAT,I0BJ,IRGWP,
1132. 2      IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
1133. 3      NETA,NLELEM,NLETA,NUELEM,NUEA,JH(50),KINBAS(122),LA(122),
1134. 4      LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
1135.  C
1136.      DO 1000 I=1,NROW
1137. 1000 X(I) = X(I) - Y(I)*DP
1138.      IF (NPIVOT .EQ. 1) GO TO 2000
1139.      KINBAS(JCOLP) = -(KINBAS(JCOLP) + 1)
1140.      IVOUT = JCOLP
1141.      RETURN
1142. 2000 X(IRGWP) = DE
1143.      IVOUT = JH(IRGWP)
1144.      KINBAS(JCOLP) = IRGWP
1145.      KINBAS(IVOUT) = IPTYPE
1146.      JH(IRGWP) = JCOLP
1147.      RETURN
1148.      END
1149.      SUBROUTINE NORMAL(ITSINV)
1150.
1151.  C
1152.  C      SERVES AS MASTER PROGRAM FOR LINEAR PROGRAMMING COMPONENT
1153.  C      (REVISED, PRIMAL-SIMPLEX METHOD) OF BRANCH-AND-BOUND ROUTINE BB
1154.  C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
1155.  C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1156.  C*****DESCRIPTION OF PARAMETERS*****
1157.  C      ITSINV = NUMBER OF SIMPLEX ITERATIONS SINCE LAST BASIS
1158.  C      INVERSION (INPUT/OUTPUT)
1159.  C*****

```



```

1159.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1160.      1      INTEGER*4 (I-N,Q)
1161.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1162.      INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
1163.      DOUBLE PRECISION E(4000)
1164.      REAL A(2000)
1165.      C
1166.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRD,QMA,QBA,QFI,
1167.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1168.      2      QE,QH,QL,QD,QR,QM,QG,NTMAX
1169.      C
1170.      COMMON/BLOCK2/ DFPART(60),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1171.      1      IPART(122),INCUMB(122),IVBND(500),IVID(500),
1172.      2      IOBND(500),NPivot,IPTYPE,RCOST,IFEAS
1173.      C
1174.      COMMON DSUM,DPRBD,CY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
1175.      1      SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,I0BJ,IROWP,
1176.      2      IVIN,IVOUT,ITCNT,INVFRQ,ITFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
1177.      3      NETA,NLELEM,NLETA,NUELEM,NUETA,JH(60),KINBAS(122),LA(122),
1178.      4      LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
1179.      C
1180.      IF (ITSINV .LT. INVFRQ) GO TO 1500
1181.      1000 CALL INVERT
1182.      ITSINV = 0
1183.      C
1184.      C      SIMPLEX CYCLE
1185.      C
1186.      1500 CALL FORMC
1187.      CALL PTRAN
1188.      CALL PRICE
1189.      IF (JCOLP .GT. 0) GO TO 3000
1190.      IF (MSTAT .EQ. Q1) GO TO 2000
1191.      MSTAT = QBL
1192.      GO TO 6000
1193.      2000 MSTAT = QN
1194.      GO TO 6000
1195.      3000 IVIN = JCOLP
1196.      CALL UNPACK(JCOLP)
1197.      CALL FTRAN(1)
1198.      CALL CHUZR
1199.      CALL LPBETA
1200.      ITCNT = ITCNT + 1
1201.      ITSINV = ITSINV + 1
1202.      IF (NPivot .EQ. 0) GO TO 4010
1203.      IF (NELEM .GT. 5600) GO TO 1000
1204.      CALL WRETA
1205.      4010 IF (ITSINV .GE. INVFRQ) GO TO 1000
1206.      IF (ITCNT .GE. ITFRQ) GO TO 6000
1207.      GO TO 1500
1208.      C
1209.      6000 RETURN
1210.      END
1211.      SUBROUTINE BANDB(INITBD,ITSINV)
1212.      C
1213.      C      MASTER PROGRAM FOR BRANCH-AND-BOUND INTEGER PROGRAMMING
1214.      C      ROUTINE. ALSO SERVES AS MASTER PROGRAM FOR REOPTIMIZATION
1215.      C      VIA REVISED DUAL-SIMPLEX METHOD AFTER A FORWARD BRANCH.
1216.      C*****DESCRIPTION OF PARAMETERS*****
1217.      C      INITBD = INITIAL LOWER BOUND ON MAXIMAL OBJECTIVE VALUE (INPJ)
1218.      C      ITSINV = NUMBER OF SIMPLEX ITERATIONS SINCE LAST BASIS
1219.      C      INVERSION (INPUT/OUTPUT)

```

```

1220. C*****
1221. C
1222.     IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1223.     1     INTEGER*4 (I-N,Q)
1224.     INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1225.     INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
1226.     DOUBLE PRECISION E(4000)
1227.     REAL A(2000)
1228.     COMMON/BLUCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1229.     1     QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1230.     2     QE,QH,QL,QQ,QR,QM,QG,NTMAX
1231.     COMMON/BLUCK2/ DFPART(60),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1232.     1     IPART(122),INCUMB(122),IVBND(500),IVID(500),
1233.     2     ICBND(500),NPIVOT,IPTYPE,RCGST,IFEAS
1234.     COMMON DSUM,DPROD,DY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
1235.     1     SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,IOBJ,IROWP,
1236.     2     IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
1237.     3     NETA,NLELEM,NLETA,NUELEM,NUETA,JH(60),KINBAS(122),LA(122),
1238.     4     LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
1239.     IF (ITCNT .GE. ITRFRQ) RETURN
1240.     IFEAS = 0
1241.     LISTL = 0
1242.     INCVAL = INITBD
1243. C     TEST FOR FATHOMING
1244.     100 CALL TESTX
1245.     IF (MSTAT .EQ. QBL) GO TO 200
1246. C     CURRENT NODE FATHOMED; BACKTRACK TO LAST PROMISING NODE ON LIST
1247.     150 CALL BKTRAK
1248. C     IF LIST IS EMPTY, RETURN TO MAIN(COMPUTATIONS COMPLETED)
1249.     IF (LISTL .EQ. 0) RETURN
1250. C     USE PRIMAL SIMPLEX METHOD FOR REOPTIMIZATION AT NEW NODE
1251.     CALL NORMAL(ITSINV)
1252.     IF (ITCNT .GE. ITRFRQ) RETURN
1253.     GO TO 100
1254. C     CURRENT NODE NOT FATHOMED; COMPUTE PENALTIES
1255. C     BRANCHING AT CURRENT NODE IS DONE FROM SUBROUTINE PENLTS
1256.     200 CALL PENLTS
1257.     IF (ICIF) 400,150,400
1258. C
1259. C     REINVERT CURRENT BASIS
1260.     1000 CALL INVERT
1261.     ITSINV = 0
1262. C
1263. C     DUAL SIMPLEX CYCLE
1264. C
1265. C     CHOOSE PIVOT ROW IROWP
1266.     300 CALL DCHUZR
1267.     IF (IROWP .GT. 0) GO TO 400
1268.     MSTAT = QBL
1269.     GO TO 100
1270. C     CHOOSE PIVOT COLUMN JCOLP
1271.     400 CALL DCHUZR
1272.     IF (JCOLP .EQ. 0) GO TO 150
1273.     IVIN = JCOLP
1274. C     UPDATE RIGHT-HAND SIDES TO REFLECT NEW BASIS RESULTING FROM
1275. C     CURRENT SIMPLEX PIVOT
1276.     CALL UPBETA
1277.     ITCNT = ITCNT + 1
1278.     IF (ITCNT .GE. ITRFRQ) RETURN
1279.     ITSINV = ITSINV + 1
1280. C     DECIDE WHETHER TO REINVERT CURRENT BASIS

```

```

1281.      IF ((NELEM .GT. 5000) .OR. (ITSINV .GE. INVFRQ)) GO TO 1000
1282.      C      REINVERSION NOT NECESSARY YET; WRITE OUT NEW ETA-VECTOR FOR
1283.      C      CURRENT SIMPLEX PIVOT
1284.      CALL WRETA
1285.      GO TO 300
1286.      END
1287.      SUBROUTINE DCHUZR
1288.      C
1289.      C      SELECTS PIVOT ROW IROWP FOR CURRENT DUAL-SIMPLEX ITERATION.
1290.      C      SETS IROWP=0 IF CURRENT BASIS IS OPTIMAL. OTHERWISE, IROWP IS
1291.      C      CHOSEN TO BE THE ROW WITH GREATEST PRIMAL INFEASIBILITY.
1292.      C
1293.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1294.      1      INTEGER*4 (I-N,Q)
1295.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1296.      INTEGER*2 IPART,INCUMB,IVBND,IVID,I0BND
1297.      DOUBLE PRECISION E(4000)
1298.      REAL A(2000)
1299.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRD,QMA,QBA,QFI,
1300.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1301.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
1302.      COMMON/BLOCK2/ DFPART(60),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1303.      1      IPART(122),INCUMB(122),IVBND(500),IVID(500),
1304.      2      I0BND(500),NPIVOT,IPTYPE,RCOST,IFEAS
1305.      COMMON DPROD,DY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
1306.      1      SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,I0BJ,IROWP,
1307.      2      IVIN,IVOUT,ITCNT,INVFRQ,ITFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
1308.      3      NETA,NLELEM,NLETA,NUELEM,NUETA,JH(60),KINBAS(122),LA(122),
1309.      4      LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
1310.      IROWP = 0
1311.      DP = -1.E10
1312.      DO 1000 I=1,NROW
1313.      IF (I .EQ. I0BJ) GO TO 1000
1314.      ICOL = JH(I)
1315.      IF (X(I) .LT. (XLB(ICOL) - ZTOLZE)) GO TO 100
1316.      IF (X(I) .GT. (XUB(ICOL) + ZTOLZE)) GO TO 200
1317.      GO TO 1000
1318.      C
1319.      C      BASIC VARIABLE ON ROW I FALLS BELOW ITS LOWER BOUND
1320.      100 DE = XLB(ICOL) - X(I)
1321.      IF (DE .LE. DP) GO TO 1000
1322.      IPTYPE = 0
1323.      GO TO 250
1324.      C
1325.      C      BASIC VARIABLE ON ROW I EXCEEDS ITS UPPER BOUND
1326.      200 DE = X(I) - XUB(ICOL)
1327.      IF (DE .LE. DP) GO TO 1000
1328.      IPTYPE = -1
1329.      C
1330.      250 IROWP = I
1331.      1000 CONTINUE
1332.      RETURN
1333.      END
1334.      SUBROUTINE DCHUZE
1335.      C
1336.      C      SELECTS PIVOT COLUMN JCOLP FOR CURRENT DUAL-SIMPLEX ITERATION.
1337.      C      SETS JCOLP=0 IF LP-PROBLEM AT CURRENT NODE IS INFEASIBLE.
1338.      C      OTHERWISE CHOOSES JCOLP TO MAINTAIN PRIMAL-OPTIMALITY.
1339.      C
1340.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1341.      1      INTEGER*4 (I-N,Q)

```



```

1342.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1343.      INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
1344.      DOUBLE PRECISION E(4000)
1345.      REAL A(2000)
1346.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRD,QMA,QBA,QF1,
1347.      1 QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1348.      2 QE,QH,QL,QO,QR,QM,QG,NTMAX
1349.      COMMON/BLOCK2/ DEPART(60),REVBND,INCVL,ICOL,LISTL,IVAL,IDIR,
1350.      1 IPART(122),INCUMB(122),IVBND(500),IVID(500),
1351.      2 IOBND(500),NPIVOT,IPTYPE,RCOST,IFEAS
1352.      COMMON DSUM,DPROD,DY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
1353.      1 SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,IOBJ,IROWP,
1354.      2 IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
1355.      3 NETA,NLELEM,NLETA,NUELEM,NUETA,JH(60),KINBAS(122),LA(122),
1356.      4 LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
1357.      C
1358.      JCOLP = 0
1359.      IF (IPTYPE .EQ. -1) GO TO 1000
1360.      C
1361.      C          LEAVING VARIABLE FALLS BELOW ITS LOWER BOUND
1362.      C
1363.      DP = -1.E10
1364.      DO 500 J=1,NCOL
1365.      IF (KINBAS(J) .GT. 0) GO TO 500
1366.      IF ((XUB(J) - XLB(J)) .LE. ZTOLZE) GO TO 500
1367.      CALL UNPACK(J)
1368.      CALL FTRAN(1)
1369.      IF (KINBAS(J) .EQ. -1) GO TO 200
1370.      IF (Y(IROWP) + ZTOLPV) 225,225,500
1371.      200 IF (Y(IROWP) - ZTOLPV) 500,225,225
1372.      C
1373.      225 DE = Y(IOBJ)/Y(IROWP)
1374.      IF (DE - DP) 500,500,250
1375.      250 JCOLP = J
1376.      DP = DE
1377.      500 CONTINUE
1378.      C
1379.      IF (JCOLP .EQ. 0) RETURN
1380.      CALL UNPACK(JCOLP)
1381.      CALL FTRAN(1)
1382.      ICOL = JH(IROWP)
1383.      DP = (X(IROWP) - XLB(ICOL))/Y(IROWP)
1384.      GO TO 2000
1385.      C
1386.      C          LEAVING VARIABLE EXCEEDS ITS UPPER BOUND
1387.      C
1388.      1000 DP = 1.E10
1389.      DO 1500 J=1,NCOL
1390.      IF (KINBAS(J) .GT. 0) GO TO 1500
1391.      IF ((XUB(J) - XLB(J)) .LE. ZTOLZE) GO TO 1500
1392.      CALL UNPACK(J)
1393.      CALL FTRAN(1)
1394.      IF (KINBAS(J) .EQ. -1) GO TO 1200
1395.      IF (Y(IROWP) - ZTOLPV) 1500,1225,1225
1396.      1200 IF (Y(IROWP) + ZTOLPV) 1225,1225,1500
1397.      C
1398.      1225 DE = Y(IOBJ)/Y(IROWP)
1399.      IF (DE - DP) 1250,1500,1500
1400.      1250 JCOLP = J
1401.      DP = DE
1402.      1500 CONTINUE

```

```

1403. C
1404. IF (JCOLP .EQ. 0) RETURN
1405. CALL UNPACK(JCOLP)
1406. CALL FTRAN(1)
1407. ICOL = JH(IROWP)
1408. DP = (X(IROWP) - XUB(ICOL))/Y(IROWP)
1409. C
1410. 2000 IF (KINBAS(JCOLP) .EQ. 0) DE = DP + XLB(JCOLP)
1411. IF (KINBAS(JCOLP) .EQ. -1) DE = DP + XUB(JCOLP)
1412. NPIVOT = 1
1413. RETURN
1414. END
1415. SUBROUTINE TESTX
1416. C
1417. C TESTS LP-OPTIMAL SOLUTION AT CURRENT NODE FOR FATHOMING.
1418. C FATHOMING OCCURS IF
1419. C (1) LP PROBLEM AT CURRENT NODE IS INFEASIBLE (MSTAT = QN); OR
1420. C (2) LP-OPTIMAL OBJ. VALUE (IVAL) .LE. OBJ. VALUE OF CURRENT
1421. C INCUMBENT SOLUTION(INCVAL); OR
1422. C (3) LP-OPTIMAL SOLUTION SATISFIES INTEGER RESTRICTIONS.
1423. C
1424. IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1425. 1 INTEGER*4 (I-N,Q)
1426. INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1427. INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
1428. DOUBLE PRECISION E(4000)
1429. REAL A(2000)
1430. COMMON/BLOCK1/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1431. 1 QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1432. 2 QE,QH,QL,QO,QR,QM,QG,NTMAX
1433. COMMON/BLOCK2/ DFPART(60),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1434. 1 IPART(122),INCUMB(122),IVBND(500),IVID(500),
1435. 2 IOBND(500),NPIVOT,IPTYPE,RCOST,IFEAS
1436. COMMON/DSUM,DPROG,DY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
1437. 1 SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,IOBJ,IROWP,
1438. 2 IVIN,IVOUT,ITCNT,INVERQ,ITRFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
1439. 3 NETA,NLELEM,NLETA,NUELEM,NUETA,JH(60),KINBAS(122),LA(122),
1440. 4 LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
1441. COMMON/TIMERS/ TFIRST,TUPT,IFVAL,ITCT
1442. DP = X(IOBJ) + ZTOLZE
1443. IVAL = IDINT(DP)
1444. IF (DP .LT. 0.) IVAL = IVAL - 1
1445. IF ((MSTAT .EQ. QN) .OR. (IVAL .LE. INCVAL)) GO TO 2000
1446. C
1447. C COMPUTE INTEGER AND FRACTIONAL PARTS OF EACH BASIC VAR.
1448. C
1449. DO 100 I=1,NROW
1450. IPART(I) = IDINT(X(I) + ZTOLZE)
1451. NTEMP(I) = IPART(I)
1452. 100 DFPART(I) = X(I) - FLOAT(NTEMP(I))
1453. C
1454. C CHECK FOR ALL-INTEGER SOLUTION
1455. C
1456. DO 200 I=1,NROW
1457. IF (JH(I) .LE. NROW) GO TO 200
1458. IF (DFPART(I) .GE. ZTOLZE) RETURN
1459. 200 CONTINUE
1460. C
1461. C SOLUTION ALL-INTEGER: INSTALL AS NEW INCUMBENT
1462. C
1463. C COMPUTE AND OUTPUT COMPUTATION TIME REQUIRED TO REACH NEW

```

```

1464. C      INCUMBENT SOLUTION.  ALSO OUTPUT OBJ. VAL. FOR THIS SOLUTION.
1465.      CALL PCLOCK(JTIME, ITOT)
1466.      TOPT = JTIME/100.
1467.      WRITE (6,1) TOPT,IVAL
1468.      1 FORMAT (' TIME =',F7.2,' SECONDS:  INCVAL =',I10)
1469.      IF (IFEAS .EQ. 1) GO TO 500
1470.      TFIRST = TOPT
1471.      IFVAL = IVAL
1472.      IFEAS = 1
1473.      500 INCVAL = IVAL
1474.      DO 1000 J=1,NCOL
1475.      IF (KINBAS(J)) 600,700,1000
1476.      600 INCUMB(J) = IDINT(XUB(J))
1477.      GO TO 1000
1478.      700 INCUMB(J) = IDINT(XLB(J))
1479.      1000 CONTINUE
1480. C
1481.      DO 1100 I=1,NROW
1482.      ICOL = JH(I)
1483.      INCUMB(ICOL) = IPART(I)
1484.      1100 CONTINUE
1485. C
1486. C      CURRENT PROBLEM NO LONGER OF INTEREST
1487. C
1488.      2000 MSTAT = QI
1489.      RETURN
1490.      END
1491.      SUBROUTINE PENLTS
1492. C
1493. C      COMPUTES TOMLIN'S IMPROVED UP- AND DOWN- PENALTIES AND THE
1494. C      GOMORY PENALTY FOR EACH NONINTEGER BASIC VARIABLE.  ALSO CHECKS
1495. C      FOR FORCED BRANCHES ON BOTH BASIC AND NONBASIC VARIABLES.  IN
1496. C      THE ABSENCE OF FORCED BRANCHES ON BASIC VARIABLES, THE
1497. C      BRANCHING VARIABLE IS CHOSEN TO BE THE ONE WITH LARGEST
1498. C      ASSOCIATED UP- OR DOWN-PENALTY.  THE FORWARD BRANCH IS TAKEN IN
1499. C      THE DIRECTION OPPOSITE TO THE MAXIMUM PENALTY.  THE NODE
1500. C      CORRESPONDING TO THE BRANCH IN THE SAME DIRECTION AS THE
1501. C      MAXIMUM PENALTY IS ADDED TO THE LIST, TO BE EXAMINED LATER.
1502. C      THE BRANCHING PROCESS ITSELF IS CARRIED OUT IN SUBROUTINE
1503. C      BRANCH, WHICH IS CALLED FROM SUBROUTINE PENLTS.
1504. C
1505.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1506.      1      INTEGER*4 (I-N,Q)
1507.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1508.      INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
1509.      DOUBLE PRECISION E(4000)
1510.      REAL A(2000)
1511.      REAL PU(60),PD(60),PG(60)
1512.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QF1,
1513.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QL,
1514.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
1515.      COMMON/BLOCK2/ DFPART(60),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1516.      1      IPART(122),INCUMB(122),IVBND(500),IVID(500),
1517.      2      IOBND(500),NPIVOT,IP TYPE,RCOST,IFEAS
1518.      COMMON DSUM,DPROD,CY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
1519.      1      SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,IOBJ,IRUWP,
1520.      2      IVIN,IROUT,ITCNT,INFRQ,ITRFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
1521.      3      NETA,NLELEM,NLETA,NUELEM,NUE TA,JH(60),KINBAS(122),LA(122),
1522.      4      LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
1523.      DO 10 I=1,NROW
1524.      IF (DFPART(I) .LT. ZTOLZE) GO TO 5

```



```

1525.          PU(I) = 1.E6
1526.          PD(I) = 1.E6
1527.          PG(I) = 1.E6
1528.          GO TO 10
1529.      5 PU(I) = 0.
1530.          PD(I) = 0.
1531.          PG(I) = 0.
1532.      10 CONTINUE
1533.  C
1534.          DO 1000 J=1,NCOL
1535.          IF (KINBAS(J) .GT. 0) GO TO 1000
1536.          IF ((XUB(J) - XLB(J)) .LE. ZTOLZE) GO TO 1000
1537.          CALL UNPACK(J)
1538.          CALL FTRAN(1)
1539.          IF (KINBAS(J) .EQ. 0) GO TO 30
1540.          DO 20 I=1,NROW
1541.      20 Y(I) = -Y(I)
1542.  C
1543.      C          CHECK FOR FORCED BRANCH ON XJ
1544.      30 IF (J .LE. NROW) GO TO 50
1545.          DP = X(IOBJ) - Y(IOBJ) + ZTOLZE
1546.          IVAL = IDINT(DP)
1547.          IF (DP .LT. 0.) IVAL = IVAL - 1
1548.          IF (IVAL .GT. INCVAL) GO TO 50
1549.          IDIR = 2*KINBAS(J) + 1
1550.          IF (IDIR .EQ. -1) REVBND = SNGL(XUB(J))
1551.          IF (IDIR .EQ. 1) REVBND = SNGL(XLB(J))
1552.          ICOL = J
1553.          CALL BRANCH
1554.          GO TO 1000
1555.  C
1556.      50 DO 500 I=1,NROW
1557.          IF (JF(I) .LE. NROW) GO TO 500
1558.          IF (DFPART(I) .LT. ZTOLZE) GO TO 500
1559.  C
1560.      C          COMPUTE UP PENALTY FOR X(I), XJ
1561.      100 IF (Y(I) .GT. -ZTOLPV) GO TO 200
1562.          DE = Y(IOBJ)*(DFPART(I) - 1.)/Y(I)
1563.          IF (DE .LT. Y(IOBJ)) DE = Y(IOBJ)
1564.          IF (DE .LT. PU(I)) PU(I) = DE
1565.          GO TO 300
1566.  C
1567.      C          COMPUTE DOWN PENALTY FOR X(I), XJ
1568.      200 IF (Y(I) .LT. ZTOLPV) GO TO 300
1569.          DE = Y(IOBJ)*DFPART(I)/Y(I)
1570.          IF (DE .LT. Y(IOBJ)) DE = Y(IOBJ)
1571.          IF (DE .LT. PD(I)) PD(I) = DE
1572.  C
1573.      C          COMPUTE GOMCRY PENALTY FOR X(I), XJ
1574.      300 DP = CABS(Y(I))
1575.          NTEMP(1) = IDINT(DP)
1576.          DP = DP - FLOAT(NTEMP(1))
1577.          IF ((DP .LE. ZTOLZE) .OR. (DP .GE. 1.-ZTOLZE)) GO TO 500
1578.          IF (Y(I) .LT. 0.) DP = 1. - DP
1579.          IF (DP .GT. DFPART(I)) GO TO 310
1580.          DE = Y(IOBJ)*DFPART(I)/DP
1581.          GO TO 320
1582.      310 DE = Y(IOBJ)*(1. - DFPART(I))/(1. - DP)
1583.      320 IF (DE .LT. PG(I)) PG(I) = DE
1584.  C
1585.      500 CONTINUE

```

```

1586.      1000 CONTINUE
1587.      C
1588.      C      COMPUTE LARGEST GOMORY PENALTY AND TEST FOR FATHOMING
1589.      PEN = 0.
1590.      DO 2000 I=1,NROW
1591.      IF (JF(I) .LE. NROW) GO TO 2000
1592.      IF (PG(I) .GT. PEN) PEN = PG(I)
1593.      2000 CONTINUE
1594.      DP = X(I0BJ) - PEN + ZTOLZE
1595.      IVAL = IDINT(DP)
1596.      IF (DP .LT. 0.) IVAL = IVAL - 1
1597.      IF (IVAL .GT. INCVAL) GO TO 3000
1598.      IDIR = 0
1599.      RETURN
1600.      C
1601.      C      PROBLEM NOT FATHOMED: CHECK FOR FORCED BRANCHES ON X(I)
1602.      3000 NTEMP(2) = 0
1603.      DO 3900 I=1,NROW
1604.      IF (JF(I) .LE. NROW) GO TO 3900
1605.      IF (PU(I) .GT. PD(I)) GO TO 3600
1606.      DP = X(I0BJ) - PD(I) + ZTOLZE
1607.      NTEMP(1) = IDINT(DP)
1608.      IF (DP .LT. 0.) NTEMP(1) = NTEMP(1) - 1
1609.      IF (NTEMP(1) .GT. INCVAL) GO TO 3900
1610.      C      FORCED BRANCH UP ON X(I)
1611.      IVAL = NTEMP(1)
1612.      IDIR = -1
1613.      NTEMP(1) = IPART(I) + 1
1614.      GO TO 3700
1615.      C
1616.      3600 DP = X(I0BJ) - PU(I) + ZTOLZE
1617.      NTEMP(1) = IDINT(DP)
1618.      IF (DP .LT. 0.) NTEMP(1) = NTEMP(1) - 1
1619.      IF (NTEMP(1) .GT. INCVAL) GO TO 3900
1620.      C      FORCED BRANCH DOWN ON X(I)
1621.      IVAL = NTEMP(1)
1622.      IDIR = 1
1623.      NTEMP(1) = IPART(I)
1624.      3700 IROWP = I
1625.      ICOL = JH(IROWP)
1626.      REVBND = FLOAT(NTEMP(1))
1627.      NTEMP(2) = 1
1628.      CALL BRANCH
1629.      3900 CONTINUE
1630.      IF (NTEMP(2) .GT. 0) GO TO 5000
1631.      C
1632.      C      NO FORCED BRANCHES: CHOOSE BRANCHING VAR. AND DIRECTION
1633.      PEN = 0.
1634.      IROWP = 0
1635.      C      DETERMINE BASIC VAR. X(IROWP) WITH MAX. UP- OR DOWN-PENALTY
1636.      DO 4900 I=1,NROW
1637.      IF (JF(I) .LE. NROW) GO TO 4900
1638.      IF (PU(I) .GT. PD(I)) GO TO 4600
1639.      IF (PD(I) .LE. PEN) GO TO 4900
1640.      PEN = PD(I)
1641.      IROWP = I
1642.      IDIR = -1
1643.      NTEMP(1) = IPART(I) + 1
1644.      REVBND = FLOAT(NTEMP(1))
1645.      GO TO 4900
1646.      4600 IF (PU(I) .LE. PEN) GO TO 4900

```

```

1647.      PEN = PU(1)
1648.      IROWP = 1
1649.      IDIR = 1
1650.      NTEMP(1) = IPART(1)
1651.      REVBND = FLOAT(NTEMP(1))
1652.      4900 CONTINUE
1653.      IF (IROWP .GT. 0) GO TO 4950
1654.      C      EACH UP- AND DOWN-PENALTY .EQ. 0. (DUAL-DEGENERACY) CHOOSE ANY
1655.      C      NONINTEGER BASIC VAR. AS BRANCHING VAR.
1656.      DO 4910 IROWP=1,NKCW
1657.      IF (JH(IROWP) .LE. NROW) GO TO 4910
1658.      IF (DEPART(IROWP) .GE. ZTOLZE) GO TO 4920
1659.      4910 CONTINUE
1660.      4920 PEN = PU(IROWP)
1661.      IDIR = 1
1662.      NTEMP(1) = IPART(IROWP)
1663.      REVBND = FLOAT(NTEMP(1))
1664.      4950 ICOL = JH(IROWP)
1665.      OP = X(10BJ) - PEN + ZTOLZE
1666.      NTEMP(1) = IDINT(OP)
1667.      IF (OP .LT. 0.) NTEMP(1) = NTEMP(1) - 1
1668.      IF (IVAL .GT. NTEMP(1)) IVAL = NTEMP(1)
1669.      C      BRANCH ON CHOSEN VARIABLE
1670.      CALL BRANCH
1671.      5000 IF (IDIR .EQ. -1) IPTYPE = 0
1672.      IF (IDIR .EQ. 1) IPTYPE = -1
1673.      RETURN
1674.      END
1675.      SUBROUTINE BRANCH
1676.      C
1677.      C      BRANCHES ON VARIABLE X(ICOL) AS DETERMINED IN SUBROUTINE PENLTS
1678.      C
1679.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1680.      1      INTEGER*4 (I-N,Q)
1681.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1682.      INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
1683.      DOUBLE PRECISION E(4000)
1684.      REAL A(2000)
1685.      COMMON/BLUCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRD,QMA,QBA,QFI,
1686.      1      QED,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1687.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
1688.      COMMON/BLUCK2/ DEPART(60),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1689.      1      IPART(122),INCUMB(122),IVBND(500),IVID(500),
1690.      2      IOBND(500),NPIVOT,IPTYPE,RCOST,IFEAS
1691.      COMMON DSUM,DPROD,DY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
1692.      1      SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,IOBJ,IROWP,
1693.      2      IVIN,IVOUT,ITCNT,INVERQ,ITFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
1694.      3      NETA,NLELEM,NLETA,NUELEM,NUETA,JH(60),KINBAS(122),LA(122),
1695.      4      LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
1696.      C
1697.      C      ICOL INDEXES BRANCHING VARIABLE CHOSEN
1698.      C      IDIR INDICATES BRANCHING DIRECTION CHOSEN
1699.      C
1700.      C      ADD OPPOSITE DIRECTION TO LIST
1701.      LISTL = LISTL + 1
1702.      IF (IDIR .EQ. -1) IVBND(LISTL) = IDINT(XLB(ICOL) + ZTOLZE)
1703.      IF (IDIR .EQ. 1) IVBND(LISTL) = IDINT(XUB(ICOL) + ZTOLZE)
1704.      IVID(LISTL) = IDIR*ICOL
1705.      IOBND(LISTL) = IVAL
1706.      C
1707.      C      REVISE BOUNDS ON BRANCHING VARIABLE FOR FORWARD DIRECTION

```



```

1708.      IF (ICIR .EQ. -1) XLB(ICOL) = DBLE(REVBND)
1709.      IF (ICIR .EQ. 1) XUB(ICOL) = DBLE(REVBND)
1710.      RETURN
1711.      END
1712.      SUBROUTINE BKTRAK
1713.      C
1714.      C      BACKTRACKS TO SELECT A PROMISING (UNFATHOMED) NODE FROM THE
1715.      C      LIST OF STORED NODES. LAST-IN-FIRST-OUT (LIFO) SELECTION RULE
1716.      C      IS EMPLOYED.
1717.      C
1718.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1719.      1      INTEGER*4 (I-N,Q)
1720.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1721.      INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
1722.      DOUBLE PRECISION E(4000)
1723.      REAL A(2000)
1724.      COMMON/BLUCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1725.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1726.      2      QE,QH,QL,QU,QR,QM,QG,NTMAX
1727.      COMMON/BLUCK2/ DFPART(60),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1728.      1      IPART(122),INCUMB(122),IVBND(500),IVID(500),
1729.      2      IOBND(500),NPIVOT,IPTYPE,RCOST,IFEAS
1730.      COMMON DSUM,DPROD,CY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
1731.      1      SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,IOBJ,IROWP,
1732.      2      IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
1733.      3      NETA,NLELEM,NLETA,NUELEM,NUE TA,JH(60),KINBAS(122),LA(122),
1734.      4      LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
1735.      NTEMP(3) = 0
1736.      C      IF LIST IS EMPTY, RETURN (COMPUTATIONS COMPLETED)
1737.      50 IF (LISTL .EQ. 0) RETURN
1738.      IF (ICBND(LISTL) .LE. INCVAL) GO TO 2000
1739.      C
1740.      C      GET NEXT NODE FROM LIST
1741.      ICOL = IVID(LISTL)
1742.      IF (ICOL .LT. 0) GO TO 100
1743.      C
1744.      NTEMP(1) = IDINT(XLB(ICOL) + ZTOLZE)
1745.      NTEMP(2) = IVBND(LISTL)
1746.      XLB(ICOL) = XUB(ICOL) + 1.
1747.      XUB(ICOL) = FLOAT(NTEMP(2))
1748.      IF (KINBAS(ICOL) .GT. 0) GO TO 1000
1749.      KINBAS(ICOL) = 0
1750.      NTEMP(3) = 1
1751.      GO TO 1000
1752.      C
1753.      100 ICOL = -ICOL
1754.      NTEMP(1) = IDINT(XUB(ICOL) + ZTOLZE)
1755.      NTEMP(2) = IVBND(LISTL)
1756.      XUB(ICOL) = XLB(ICOL) - 1.
1757.      XLB(ICOL) = FLOAT(NTEMP(2))
1758.      IF (KINBAS(ICOL) .GT. 0) GO TO 1000
1759.      KINBAS(ICOL) = -1
1760.      NTEMP(3) = 1
1761.      C
1762.      1000 IVID(LISTL) = -IVID(LISTL)
1763.      IVBND(LISTL) = NTEMP(1)
1764.      IOBND(LISTL) = -10000
1765.      C
1766.      C      UPDATE X
1767.      IF (NTEMP(3) .EQ. 0) RETURN
1768.      CALL SHIFTR(1,3)

```

```

1647.      PEN = PU(I)
1648.      IROWP = I
1649.      IDIR = 1
1650.      NTEMP(1) = IPART(I)
1651.      REVBND = FLOAT(NTEMP(1))
1652.      4900 CONTINUE
1653.      IF (IROWP .GT. 0) GO TO 4950
1654.      C      EACH UP- AND DOWN-PENALTY .EQ. 0. (DUAL-DEGENERACY) CHOOSE ANY
1655.      C      NONINTEGER BASIC VAR. AS BRANCHING VAR.
1656.      DO 4910 IROWP=1,NROW
1657.      IF (JH(IROWP) .LE. NROW) GO TO 4910
1658.      IF (DEPART(IROWP) .GE. ZTOLZE) GO TO 4920
1659.      4910 CONTINUE
1660.      4920 PEN = PU(IROWP)
1661.      IDIR = 1
1662.      NTEMP(1) = IPART(IROWP)
1663.      REVBND = FLOAT(NTEMP(1))
1664.      4950 ICOL = JH(IROWP)
1665.      DP = X(10BJ) - PEN + ZTOLZE
1666.      NTEMP(1) = IDINT(DP)
1667.      IF (DP .LT. 0.) NTEMP(1) = NTEMP(1) - 1
1668.      IF (IVAL .GT. NTEMP(1)) IVAL = NTEMP(1)
1669.      C      BRANCH ON CHOSEN VARIABLE
1670.      CALL BRANCH
1671.      5000 IF (ICIR .EQ. -1) IPTYPE = 0
1672.      IF (ICIR .EQ. 1) IPTYPE = -1
1673.      RETURN
1674.      END
1675.      SUBROUTINE BRANCH
1676.      C
1677.      C      BRANCHES ON VARIABLE X(ICOL) AS DETERMINED IN SUBROUTINE PENLTS
1678.      C
1679.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1680.      1      INTEGER*4 (I-N,Q)
1681.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1682.      INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
1683.      DOUBLE PRECISION E(4000)
1684.      REAL A(2000)
1685.      COMMON/BLUCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1686.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1687.      2      QE,QH,QL,QQ,QR,QM,QG,NTMAX
1688.      COMMON/BLUCK2/ DEPART(60),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1689.      1      IPART(122),INCUMB(122),IVBND(500),IVID(500),
1690.      2      IOBND(500),NPIVOT,IPTYPE,RCOST,IFEAS
1691.      COMMON DSUM,DPROD,DY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
1692.      1      SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,IOBJ,IROWP,
1693.      2      IVIN,IVOUT,ITCNT,INVRQ,ITFRQ,IFFEZ,JCOLD,NROW,NCOL,NELEM,
1694.      3      NLETA,NLELEM,NLETA,NUELEM,NUETA,JH(60),KINBAS(122),LA(122),
1695.      4      LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
1696.      C
1697.      C      ICOL INDEXES BRANCHING VARIABLE CHOSEN
1698.      C      IDIR INDICATES BRANCHING DIRECTION CHOSEN
1699.      C
1700.      C      ADD OPPOSITE DIRECTION TO LIST
1701.      LISTL = LISTL + 1
1702.      IF (ICIR .EQ. -1) IVBND(LISTL) = IDINT(XLB(ICOL) + ZTOLZE)
1703.      IF (ICIR .EQ. 1) IVBND(LISTL) = IDINT(XUB(ICOL) + ZTOLZE)
1704.      IVID(LISTL) = IDIR*ICOL
1705.      IOBND(LISTL) = IVAL
1706.      C
1707.      C      REVISE BOUNDS ON BRANCHING VARIABLE FOR FORWARD DIRECTION

```

```

1703.      IF (ICIR .EQ. -1) XLB(ICOL) = DBLE(REVBND)
1709.      IF (ICIR .EQ. 1) XJB(ICOL) = DBLE(REVBND)
1710.      RETURN
1711.      END
1712.      SUBROUTINE BKTRAK
1713.      C
1714.      C      BACKTRACKS TO SELECT A PROMISING (UNFATHOMED) NODE FROM THE
1715.      C      LIST OF STORED NODES.  LAST-IN-FIRST-OUT (LIFO) SELECTION RULE
1716.      C      IS EMPLOYED.
1717.      C
1718.      IMPLICIT REAL**4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1719.      1      INTEGER*4 (I-N,Q)
1720.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1721.      INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
1722.      DOUBLE PRECISION E(4000)
1723.      REAL A(2000)
1724.      COMMON/BLUCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1725.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1726.      2      QE,QH,QL,QQ,QR,QM,QG,NTMAX
1727.      COMMON/BLUCK2/ DEPART(60),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1728.      1      IPART(122),INCUMB(122),IVBND(500),IVID(500),
1729.      2      ICBND(500),NPIVOT,IPTYPE,RCOST,IFEAS
1730.      COMMON DSUM,DPROD,CY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
1731.      1      SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,IQBJ,IROWP,
1732.      2      IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,IFFEZ,JCOLP,NROW,NCUL,NELEM,
1733.      3      NETA,NLELEM,NLETA,NUELEM,NUETA,JH(60),KINBAS(122),LA(122),
1734.      4      LE(1002),IA(2000),IE(4000),XLB(122),XJB(122)
1735.      NTEMP(3) = 0
1736.      C      IF LIST IS EMPTY, RETURN (COMPUTATIONS COMPLETED)
1737.      50 IF (LISTL .EQ. 0) RETURN
1738.      IF (ICBND(LISTL) .LE. INCVAL) GO TO 2000
1739.      C
1740.      C      GET NEXT NODE FROM LIST
1741.      ICOL = IVID(LISTL)
1742.      IF (ICOL .LT. 0) GO TO 100
1743.      C
1744.      NTEMP(1) = IDINT(XLB(ICOL) + ZTOLZE)
1745.      NTEMP(2) = IVBND(LISTL)
1746.      XLB(ICOL) = XJB(ICOL) + 1.
1747.      XJB(ICOL) = FLOAT(NTEMP(2))
1748.      IF (KINBAS(ICOL) .GT. 0) GO TO 1000
1749.      KINBAS(ICOL) = 0
1750.      NTEMP(3) = 1
1751.      GO TO 1000
1752.      C
1753.      100 ICOL = -ICOL
1754.      NTEMP(1) = IDINT(XJB(ICOL) + ZTOLZE)
1755.      NTEMP(2) = IVBND(LISTL)
1756.      XJB(ICOL) = XLB(ICOL) - 1.
1757.      XLB(ICOL) = FLOAT(NTEMP(2))
1758.      IF (KINBAS(ICOL) .GT. 0) GO TO 1000
1759.      KINBAS(ICOL) = -1
1760.      NTEMP(3) = 1
1761.      C
1762.      1000 IVID(LISTL) = -IVID(LISTL)
1763.      IVBND(LISTL) = NTEMP(1)
1764.      ICBND(LISTL) = -10000
1765.      C
1766.      C      UPDATE X
1767.      IF (NTEMP(3) .EQ. 0) RETURN
1768.      CALL SHIFTR(1,3)

```



```

1769.      DO 9000 J=1,NCOL
1770.      IF (KINBAS(J)) 8600,8700,9000
1771.      8600 DE = XUB(J)
1772.      GO TO 8750
1773.      8700 DE = XLB(J)
1774.      8750 LL = LA(J)
1775.      KK = LA(J+1) - 1
1776.      DO 8800 I=LL,KK
1777.      IR = IA(I)
1778.      8800 Y(IR) = Y(IR) - A(I)*DE
1779.      9000 CONTINUE
1780.      CALL FTRAN(1)
1781.      CALL SHIFTR(3,2)
1782.      RETURN
1783.
1784.      C      NODE FATHOMED: UPDATE VAR. BOUNDS AND BACKTRACK AGAIN
1785.      2000 ICOL = IVID(LISTL)
1786.      IF (ICOL .LT. 0) GO TO 2100
1787.
1788.      C      NTEMP(1) = IVBND(LISTL)
1789.      IF (KINBAS(ICOL)) 2010,2050,2050
1790.      2010 NTEMP(3) = 1
1791.      DP = XUB(ICOL) - XLB(ICOL)
1792.      DY = FLOAT(NTEMP(1)) - XUB(ICOL)
1793.      IF (DP .LT. DY) KINBAS(ICOL) = 0
1794.      2050 XJB(ICOL) = FLOAT(NTEMP(1))
1795.      GO TO 3000
1796.
1797.      C      2100 ICOL = -ICOL
1798.      NTEMP(1) = IVBND(LISTL)
1799.      IF (KINBAS(ICOL)) 2150,2110,2150
1800.      2110 NTEMP(3) = 1
1801.      DY = XLB(ICOL) - FLOAT(NTEMP(1))
1802.      DP = XUB(ICOL) - XLB(ICOL)
1803.      IF (DP .LT. DY) KINBAS(ICOL) = -1
1804.      2150 XLB(ICOL) = FLOAT(NTEMP(1))
1805.
1806.      C      3000 LISTL = LISTL - 1
1807.      GO TO 50
1808.
1809.      C      END
1810.      SUBROUTINE WRAPUP
1811.
1812.      C      OUTPUTS OPTIMAL SOLUTION AND CORRESPONDING OBJECTIVE VALUE.
1813.
1814.      C      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1815.      1      INTEGER*4 (I-N,Q)
1816.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1817.      INTEGER*2 IPART,INCUMB,IVBND,IVID,IOBND
1818.      DOUBLE PRECISION E(4000)
1819.      REAL A(2000)
1820.      COMMON/BLOCK2/ DFPART(60),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1821.      1      IPART(122),INCUMB(122),IVBND(500),IVID(500),
1822.      2      IOBND(500),NPIVOT,IPTYPE,RCOST,IFEAS
1823.      COMMON DSJH,DPROD,DY,DE,DP,B(60),X(60),Y(60),YTEMP(60),A,E,
1824.      1      SUMINF,ICNAM(122,2),NAME(20),NTEMP(20),MSTAT,IOBJ,IRDWP,
1825.      2      IVIN,IVOUT,ITCNT,INVERQ,ITFRQ,IFFEZ,JCOLP,NROW,NCOL,NELEM,
1826.      3      NETA,NLELEM,NLETA,NUELEM,NUEA,JH(60),KINBAS(122),LA(122),
1827.      4      LE(1002),IA(2000),IE(4000),XLB(122),XUB(122)
1828.      IF (ITCNT .GE. ITFRQ) GO TO 20
1829.      IF (IFEAS .EQ. 0) GO TO 10

```

```

1830.      WRITE (6,1)
1831.      1 FORMAT (///' OPTIMAL INTEGER SOLUTION X(J), J=1,...,NCOL'//)
1832.      WRITE (6,2) (INCUMB(J), J=1,NCOL)
1833.      2 FORMAT (20I5)
1834.      WRITE (6,3) INCVAL
1835.      3 FORMAT (///' MAX OBJECTIVE VALUE =', I5)
1836.      RETURN
1837.      10 WRITE (6,4) INCVAL
1838.      4 FORMAT (' NO FEASIBLE SOLUTION FOUND WITH OBJECTIVE VALUE .GT.',
1839.      1      I10)
1840.      RETURN
1841.      20 WRITE (6,100) ITCNT
1842.      100 FORMAT (' SIMPLEX ITERATIONS =',I3,' : COMPUTATIONS TERMINATED.')
1843.      IF (IFEAS .EQ. 0) GO TO 10
1844.      WRITE (6,101)
1845.      101 FORMAT (///' BEST INTEGER SOLUTION FOUND IS'//)
1846.      WRITE (6,2) (INCUMB(J), J=1,NCOL)
1847.      RETURN
1848.      END

```

\*\*\*\*\*SAMPLE INPUT DATA FOR BRANCH-AND-BOUND CODE BB\*\*\*\*\*

I 1 10999 0  
NAME SAMPLE

ROWS

N 08J  
L ROW1  
L ROW11  
L ROW12  
L ROW13  
L ROW14  
L ROW21  
L ROW22  
L ROW23  
L ROW24  
L ROW31  
L ROW32  
L ROW33  
L ROW34  
L ROW41  
L ROW42  
L ROW43  
L ROW44

COLUMNS

COL11	CBJ	-50.	ROW1	40.
COL11	RCW11	16.	ROW12	38.
COL11	RCW13	8.	ROW14	38.
COL12	CBJ	-113.	ROW1	91.
COL12	RCW11	92.	ROW12	39.
COL12	RCW13	71.	ROW14	52.
COL13	CBJ	-30.	ROW1	10.
COL13	RCW11	41.	ROW12	32.
COL13	RCW13	30.	ROW14	30.
COL14	CBJ	-62.	ROW1	30.
COL14	RCW11	10.	ROW12	71.
COL14	RCW13	60.	ROW14	42.
COL15	CBJ	-210.	ROW1	160.
COL15	RCW11	150.	ROW12	80.
COL15	RCW13	200.	ROW14	170.
COL21	CBJ	-43.	ROW1	20.
COL21	RCW21	23.	ROW22	20.
COL21	RCW23	18.	ROW24	9.
COL22	CBJ	-7.	ROW1	3.
COL22	RCW21	4.	ROW22	5.
COL22	RCW23	0.	ROW24	7.
COL23	CBJ	-33.	ROW1	12.
COL23	RCW21	16.	ROW22	40.
COL23	RCW23	30.	ROW24	20.
COL24	CBJ	-5.	ROW1	3.
COL24	RCW21	6.	ROW22	8.
COL24	RCW23	4.		
COL25	CBJ	-12.	ROW1	18.
COL25	RCW22	12.	ROW23	8.
COL25	RCW24	3.		
COL31	CBJ	-32.	ROW1	9.
COL31	RCW31	12.	ROW32	30.
COL31	RCW33	31.	ROW34	21.
COL32	CBJ	-20.	ROW1	25.



COL32	RCW31	0.		KJW32	15.
COL32	RCW33	0.		ROW34	4.
COL33	CBJ	-4.		ROW1	1.
COL33	RCW31	2.		ROW33	3.
COL33	RCW34	1.			
COL34	CBJ	-5.		ROW1	1.
COL34	RCW31	1.		ROW32	1.
COL34	RCW34	2.			
COL35	CBJ	-43.		ROW1	10.
COL35	RCW32	23.		ROW33	18.
COL35	RCW34	14.			
COL41	CBJ	-426.		ROW1	280.
COL41	RCW41	200.		ROW42	100.
COL41	RCW43	60.		ROW44	310.
COL42	CBJ	-42.		ROW1	10.
COL42	RCW41	20.		ROW43	21.
COL42	RCW44	8.			
COL43	CBJ	-12.		ROW1	8.
COL43	RCW41	0.		ROW42	20.
COL43	RCW43	4.		ROW44	4.
COL44	CBJ	-8.		ROW1	1.
COL44	RCW41	2.		ROW42	3.
COL44	RCW44	6.			
COL45	CBJ	-2.		ROW1	1.
COL45	RCW41	1.		ROW43	2.
COL45	RCW44	1.			
RHS					
RHS1	RCW1	293.			
RHS1	RCW11	221.		ROW12	182.
RHS1	RCW13	238.		ROW14	232.
RHS1	RCW21	30.		ROW22	64.
RHS1	RCW23	40.		ROW24	27.
RHS1	RCW31	10.		ROW32	48.
RHS1	RCW33	41.		ROW34	29.
RHS1	RCW41	160.		ROW42	86.
RHS1	RCW43	51.		ROW44	230.
ENDATA					
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
1.	1.	1.	1.	1.	1.
1.	1.	1.	1.	1.	1.

\*\*\*\*\*OUTPUT GENERATED BY BB ON SAMPLE PROBLEM\*\*\*\*\*

PROBLEM 1  
TIME = 2.94 SECONDS: INCVAL = 414  
TIME = 3.07 SECONDS: INCVAL = 431  
TIME = 3.53 SECONDS: INCVAL = 445  
TIME = 3.56 SECONDS: INCVAL = 462  
TOTAL SOLUTION TIME = 6.65 SECONDS

OPTIMAL INTEGER SOLUTION X(J), J=1,...,NCOL

462	21	56	2	89	70	3	13	10	8	5	9	14	8	131	63	34	211	1	1
1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	1	1	1		

MAX OBJECTIVE VALUE = 462

APPENDIX II

LISTING AND SAMPLE INPUT/OUTPUT FOR COMPUTER PROGRAM DSLC



```

1.      C
2.      C      DECOMPOSITION ROUTINE, DSLC
3.      C      DECOMPOSITION AND SOLUTION OF PURE-INTEGER LINEAR PROGRAMS
4.      C      (WITH GENERAL INTEGER VARIABLES) WITH BLOCK ANGULAR CONSTRAINT
5.      C      MATRICES.
6.      C*****SINGLE LINKING CONSTRAINT CASE*****
7.      C      GARY A. KOCHMAN (OPERATIONS RESEARCH), SEPT. 1976, MOD 1
8.      C
9.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
10.     1      INTEGER*4 (I-N,Q)
11.     1      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
12.     1      INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
13.     1      INTEGER*2 IZSTAR,IBID,IBBND,IZBND
14.     1      INTEGER*2 IAS,LAS
15.     1      DOUBLE PRECISION E(1000)
16.     1      DIMENSION A(500),INC(10),AV(10),NTSOL(10)
17.     1      COMMON/BLOCK1/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRD,QMA,QBA,QF1,
18.     1      QEQ,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
19.     1      QE,QH,QL,QO,QR,QM,QG,NTMAX
20.     1      COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
21.     1      NPIVOT,IPTYPE,RCOST,IFEAS,IPART(122),INCUMB(122),
22.     1      IVBND(500),IVID(500),ICBND(500)
23.     1      COMMON/BLOCK3/ RLAMDA(50,10),ZNAUT(10),ZJBAR(10),NSUBS,IZINC,
24.     1      IZBAR,IDUM,IBLL,NJFIX,JFIXSM,IB,JFIX(10),NSOL(10),NLAMDA(10),
25.     1      2KSEG(10),IBSTAR(200,10),IBUB(10),IBLB(10),IBP(51,10),
26.     1      3IBUSEC(200,10),IZSTAR(200,10),IBID(200),IBBND(200),IZBND(200)
27.     1      COMMON/BLOCK4/ XLBS(50,10),XUBS(50,10),AMATRS(1000),RHS(20,10),
28.     1      INROWS(10),NCOLS(10),JFIRST(11),IAS(1000),LAS(50,10)
29.     1      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
30.     1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
31.     1      IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
32.     1      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
33.     1      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
34.      C
35.      C      MAIN PROGRAM
36.      C      COORDINATES APPLICATION OF THE VARIOUS STEPS OF THE DECOMPOSI-
37.      C      TION ALGORITHM.
38.      C
39.      C      INITIALIZE PARAMETERS
40.      10 CALL FCLOCK(ITOT)
41.      READ (5,1,END=9999) NSUBS,IB,IPARM,IZINC
42.      1 FORMAT (4I10)
43.      DO 20 J=1,122
44.      ICNAM(J,1) = 0
45.      20 ICNAM(J,2) = 0
46.      IOBJ = 1
47.      ITRFRQ = 99999
48.      IBLL = 0
49.      IPAR = 0
50.      NJFIX = 0
51.      JFIXSM = 0
52.      JFIRST(1) = 1
53.      TALC = 0.
54.      TSUBS = 0.
55.      IFIRST = 0
56.      NALC = 0
57.      C
58.      C      PHASE 1
59.      DO 100 J=1,NSUBS
60.      AV(J) = 0.

```

```

61.      NTSOL(J) = 0
62.      NSOL(J) = 0
63.      JFIX(J) = 0
64.      C      INPUT DATA FOR SUBPROBLEM J
65.      CALL INPUT(J)
66.      C      SOLVE SUBPROBLEM J AS LP, PARAMETRICALLY IN THE RIGHT-HAND
67.      C      SIDE FOR THE SINGLE LINKING CONSTRAINT
68.      100 CALL PHASE1(J)
69.      WRITE (6,8)
70.      8 FORMAT ('1 START NEW PROBLEM')
71.      C      OUTPUT SLOPES AND (ALL-INTEGERS) BREAKPOINTS DEFINING THE
72.      C      SEGMENTS OF THE (MODIFIED) CONCAVE AND PIECEWISE LINEAR
73.      C      (PARAMETRIC) OBJECTIVE FUNCTIONS FOR THE SUBPROBLEMS
74.      DO 200 J=1,NSUBS
75.      LAST = NLAMDA(J) + 1
76.      WRITE (6,2) J,(IBP(K,J), K=1, LAST)
77.      2 FORMAT (// ' IBP(K, ', I3, ') = ', 20I5/25I5/5I5)
78.      LAST = LAST - 1
79.      WRITE (6,3) J,(RLAMDA(K,J), K=1, LAST)
80.      3 FORMAT (' RLAMDA(K, ', I3, ') = ', 19F6.2/20F6.2/11F6.2)
81.      200 WRITE (6,4) J,ZNAUT(J)
82.      4 FORMAT (' ZNAUT( ', I3, ') = ', F10.4)
83.      WRITE (6,5)
84.      5 FORMAT (//)
85.      CALL PCLOCK(JTIME, ITOT)
86.      TPH1 = JTIME/100.
87.      GO TO 2100
88.      C
89.      C      PHASE 2
90.      C
91.      C      BACKTRACK TO LAST PROMISING NODE IN ALLOCATION-VARIABLES SEARCH
92.      C      TREE
93.      2000 CALL BBKTRK
94.      IPAR = 0
95.      C      IF LIST OF STORED NODES IS EMPTY, COMPUTATIONS ARE COMPLETED
96.      IF (IBLL .EQ. 0) GO TO 9000
97.      2100 NALO = NALO + 1
98.      CALL PCLOCK(JTIME, ITOT)
99.      C      SOLVE MASTER PROBLEM TO OBTAIN PROMISING NEW ALLOCATION OF
100.     C      LINKING RESOURCE TO SUBPROBLEMS
101.     CALL ALOCTE(IPAR, IRHS)
102.     CALL PCLOCK(JTIME1, ITOT)
103.     TALO = TALO + (JTIME1 - JTIME)/100.
104.     IPAR = 1
105.     C      TEST FOR FATHOMING OF CURRENT NODE
106.     IF (INCVAL .LE. IZINC) GO TO 2000
107.     IZBAR = INCVAL
108.     IF (IRHS .GT. 0) IZBAR = -10000
109.     C      CHECK IF ANY SUBPROBLEM HAS PREVIOUSLY BEEN SOLVED UNDER ITS
110.     C      CURRENT ALLOCATION
111.     CALL CCMCHK(J,K)
112.     IF (ICOM .EQ. 0) GO TO 3000
113.     LAST = NSOL(J) + 1
114.     IRHS = IBSTAR(LAST, J) - IBUSED(K, J)
115.     C      BRANCH ON ALLOCATION VARIABLE IN SUBPROBLEM J
116.     2200 CALL CMBRAN(J,K)
117.     IF (MSTAT .EQ. 0) GO TO 2000
118.     C      FIX ALLOCATION TO SUBPROBLEM J AT LEVEL K
119.     2300 JFIX(J) = K
120.     NJFIX = NJFIX + 1
121.     JFIXSM = JFIXSM + IZSTAR(K, J)

```

```

122.      IF (NJFIX .LT. NSUBS) GO TO 2100
123.      C      ALL SUBS FIXED: COMPARE SOLUTION WITH INCUMBENT
124.      IF (JFIXSM .LE. IZINC) GO TO 2500
125.      DO 2400 I=1,NSUBS
126.      2400 INC(I) = JFIX(I)
127.      IZINC = JFIXSM
128.      CALL PCLOCK(JTIME,ITOT)
129.      TOPT = JTIME/100.
130.      WRITE (6,93) TOPT, IZINC
131.      93 FORMAT (' TIME =',F7.2,' SECONDS:  IZINC =',I10)
132.      IF (IFIRST .EQ. 1) GO TO 2500
133.      TFIRST = TOPT
134.      IFIRST = 1
135.      IFVAL = IZINC
136.      2500 JFIX(J) = 0
137.      JFIXSM = JFIXSM - IZSTAR(K,J)
138.      NJFIX = NJFIX - 1
139.      GO TO 2000
140.      C
141.      C      SELECT NEXT SUBPROBLEM TO SOLVE
142.      3000 AVMAX = -10000.
143.      J = 0
144.      C      FIRST TRY TO SELECT AN UNFIXED SUBPROBLEM FOR WHICH THE ALLOCA-
145.      C      TION VARIABLE IS NOT ON AN UP-BRANCH
146.      DO 3100 I=1,NSUBS
147.      IF (JFIX(I) .NE. 0) GO TO 3100
148.      IF (AVMAX .GE. AV(I)) GO TO 3100
149.      J = I
150.      AVMAX = AV(I)
151.      3100 CONTINUE
152.      IF (J .GT. 0) GO TO 3200
153.      C      ALLOCATION VARIABLE FOR EACH UNFIXED SUBPROBLEM CURRENTLY ON AN
154.      C      UP-BRANCH; SELECT UNFIXED SUBPROBLEM IN ACCORDANCE WITH
155.      C      DECISION RULE SPECIFIED BY PARAMETER IPARM ALONE
156.      DO 3150 I=1,NSUBS
157.      IF (JFIX(I) .GT. 0) GO TO 3150
158.      IF (AVMAX .GE. AV(I)) GO TO 3150
159.      J = I
160.      AVMAX = AV(I)
161.      3150 CONTINUE
162.      3200 NTSOL(J) = NTSOL(J) + 1
163.      CALL PCLOCK(JTIME,ITOT)
164.      C      LOAD PROBLEM DATA FOR SUBPROBLEM J
165.      CALL LDDATA(J)
166.      ITSINV = 99999
167.      IFEAS = 0
168.      C      GET OPTIMAL LP-SOLUTION TO SUBPROBLEM J
169.      CALL NORMAL
170.      C      TEST IF SUBPROBLEM J IS LP-FEASIBLE
171.      IF (MSTAT .EQ. QN) GO TO 3300
172.      C      SUBPROBLEM J LP-FEASIBLE; GET INITIAL LOWER BOUND ESTIMATE ON
173.      C      MAX. IP-CBJ. VAL.
174.      CALL GETLBD(J)
175.      INITBD = INCUMB(I)
176.      C      USE BRANCH-AND-BOUND ROUTINE TO COMPUTE OPT. IP-SOLUTION TO
177.      C      SUBPROBLEM J
178.      CALL EANDB(INITBD)
179.      3300 CALL PCLOCK(JTIME1,ITOT)
180.      TIME = (JTIME1 - JTIME)/100.
181.      TSUBS = TSUBS + TIME
182.      C      UPDATE SUBPROBLEM SELECTION RULE INFORMATION FOR SUBPROBLEM J,

```



```

183. C      IN ACCORDANCE WITH VALUE SPECIFIED FOR IPARM
184. GO TO (3310,3320,3330,3400),IPARM
185. 3310 AV(J)=((NTSQL(J)-1)*AV(J)+(ZJBAR(J)-INCVAL))/NTSQL(J)
186. GO TO 3400
187. 3320 AV(J) = ((NTSQL(J)-1)*AV(J) + INCUMB(2))/NTSQL(J)
188. GO TO 3400
189. 3330 AV(J) = ((NTSQL(J)-1)*AV(J) - TIME)/NTSQL(J)
190. C      SAVE ALLOCATION AND SOLUTION INFORMATION FOR SUBPROBLEM J
191. 3400 CALL SAVER(J)
192. IRHS = INCUMB(2)
193. K = NTEMP(2)
194. C      OUTPUT SOLUTION INFORMATION FOR SUBPROBLEM J
195. WRITE (6,6) J,K,(INCUMB(KK), KK=1,NCOL)
196. 6 FORMAT (' SUB',I2,' SOL',I3,' : ',2GI5)
197. C      TEST IF ALL SUBPROBLEMS HAVE BEEN SOLVED; IF SO, NO BRANCHING
198. C      NECESSARY
199. IF (NJFIX - NSUBS + 1) 2200,2300,9999
200. C
201. C      COMPUTATIONS COMPLETED; OUTPUT OPTIMAL SOLUTION INFORMATION
202. 9000 CALL CUTSOL(INC)
203. CALL PCLOCK(JTIME,ITOT)
204. TTOT = JTIME/100.
205. C      OUTPUT ALGORITHMIC PERFORMANCE INFORMATION
206. WRITE (6,7) TTOT
207. 7 FORMAT (' TOTAL SOLUTION TIME =',F7.2,' SECONDS')
208. WRITE (6,92) TPH1,NALD,TALD,(NTSQL(J), J=1,NSUBS)
209. 92 FORMAT (' TIME IN PHASE1 =',F6.2,' SECONDS',' MASTER PROBLEM SOLVE
210. 10',I10,' TIMES: TOTAL TIME SOLVING MASTER PROBLEM =',F6.2,' SECON
211. 2DS',' NUMBER OF TIMES EACH SUBPROBLEM SOLVED =',I8X,I0I5)
212. WRITE (6,9) (NSQL(J), J=1,NSUBS)
213. 9 FORMAT (' NUMBER OF DISTINCT SOLUTIONS OBTAINED IN EACH SUBPROBLEM
214. 1 =',I0I5)
215. WRITE (6,91) TSUBS
216. 91 FORMAT (' TOTAL TIME SOLVING SUBPROBLEMS =',F6.2,
217. 1' SECCNDS')
218. GO TO 10
219. 9999 STOP
220. END
221. BLOCK DATA
222. C
223. C      INITIALIZES GLOBAL PROGRAM CONSTANTS
224. C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
225. C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
226. C
227. IMPLICIT REAL*4 (A,C,E-H,O,P,R-h,Z), REAL*8 (B,D,X,Y),
228. 1 INTEGER*4 (I-N,Q)
229. COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
230. 1 QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
231. 2 QE,QH,QL,QO,QR,QM,QG,NTMAX
232. REAL ZTOLZE/1.E-5/,ZTOLPV/1.E-4/,ZTCOST/1.E-3/,ZTOLRJ/1.E-4/,
233. 1 ZTOLSM/1.E-10/
234. INTEGER*4 NRMAX/60/,NTMAX/250/,NEMAX/1000/
235. INTEGER*4 QRO/'ROW' '//,QMA/'MATR'//,QBA/'BASI'//,QFI/'FIRS'//,
236. 1QEO/'ECF' '//,QBL/' '//,QA/'A' '//,QPL/' + '//,QMI/' - '//,
237. 2QZ/' Z '//,QI/' I '//,QF/' F '//,QN/' N '//,QU/' U '//,
238. 3QB/' B '//,QC/' C '//,QE/' E '//,QH/' H '//,QL/' L '//,
239. 4QO/' O '//,QK/' R '//,QM/' M '//,QG/' G '//
240. END
241. SUBROUTINE INPUT(IFPROB)
242. C
243. C      INPUTS PROBLEM DATA

```

```

244. C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
245. C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
246. C
247. C*****DESCRIPTION OF PARAMETERS*****
248. C      IFPROB = INPROB = IDENTIFICATION NUMBER OF SUBPROBLEM BEING
249. C      INPUT (USER MUST INPUT SUBPROBLEMS IN SEQUENCE 1,2,...,NSUBS)
250. C*****
251. C
252. C      IMPLICIT REAL*4 (A,C,E-H,G,P,R-W,Z), REAL*8 (B,D,X,Y),
253. C      1      INTEGER*4 (I-N,Q)
254. C      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
255. C      INTEGER*2 IZSTAR,IBID,IBBND,IZBND
256. C      INTEGER*2 IAS,LAS
257. C      DOUBLE PRECISION E(1000),ATEMP1,ATEMP2
258. C      REAL A(500)
259. C
260. C      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
261. C      1      QEO,QBL,QA,QPL,CM1,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
262. C      2      QE,QF,QL,QG,QR,QM,QG,NTMAX
263. C
264. C      COMMON/BLOCK3/ RLAMDA(50,10),ZNAUT(10),ZJBAR(10),NSUBS,IZINC,
265. C      1      IZBAR,IDOM,IBLL,NJFIX,JFIXSM,IB,JFIX(10),NSOL(10),NLAMDA(10),
266. C      2      ZKSEG(10),IBSTAR(200,10),IBUB(10),IBLB(10),IBP(51,10),
267. C      3      IBUSED(200,10),IZSTAR(200,10),IBID(200),IBBND(200),IZBND(200)
268. C      COMMON/BLOCK4/ XLBS(50,10),XUBS(50,10),AMATRS(1000),RHS(20,10),
269. C      1      INROWS(10),NCOLS(10),JFIRST(11),IAS(1000),LAS(50,10)
270. C      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
271. C      1      OY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
272. C      2      IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVERQ,ITRFRQ,ITSINV,IFFE2,
273. C      3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
274. C      4      IE(1000),IA(500),LE(252),LAI(122),KINBAS(122),JH(60)
275. C
276. C      DO 10 I=1,60
277. C      10 B(I) = 0.
278. C      DO 20 J=1,122
279. C      20 KINBAS(J) = 0
280. C      NROW = 0
281. C      ITCNT = 0
282. C      ICS1=C
283. C      ICS2=C
284. C      READ (5,7000) INPRCB
285. C      7000 FORMAT (5I4)
286. C      IFPROB = INPROB
287. C      IF (IFPROB .EQ. 0) RETURN
288. C
289. C      5 READ(5,101) K1,K2,K3,K4,(NAME(I),I=1,4),ATEMP1,NAME(5),NAME(6),
290. C      1ATEMP2
291. C      101 FORMAT(4A1,2A4,2X,2A4,2X,F12.4,3X,2A4,2X,F12.4)
292. C      IF(K1 .EQ. QE) GO TO 600
293. C      IF(K1 .EQ. QBL) GO TO 50
294. C      IF(K1 .EQ. QN) GO TO 100
295. C      IF(K1 .EQ. QR .AND. K2 .EQ. QO) L=1
296. C      IF(K1 .EQ. QR .AND. K2 .EQ. QO) GO TO 150
297. C      IF(K1 .EQ. QC) L=2
298. C      IF(K1 .EQ. QC) GO TO 150
299. C      IF(K1 .EQ. QB .AND. K2 .EQ. QA) L=3
300. C      IF(K1 .EQ. QB .AND. K2 .EQ. QA) GO TO 150
301. C      IF(K1 .EQ. QR .AND. K2 .EQ. QH) L=4
302. C      IF(K1 .EQ. QR .AND. K2 .EQ. QH) GO TO 150
303. C      50 GO TO(210,320,410,500),L
304. C      100 NTEMP(1) = NAME(3)

```

```

305.      NTEMP(2) = NAME(4)
306.      150 CONTINUE
307.      GO TO 5
308.      C
309.      210 NROW=NROW+1
310.      NCCL=NROW
311.      ICNAM(NROW,1) = NAME(1)
312.      ICNAM(NROW,2) = NAME(2)
313.      C
314.      C      TEST ROW TYPE
315.      C
316.      IF(K2.EQ.QL .OR. K3.EQ.QL) GO TO 220
317.      IF(K2.EQ.QE .OR. K3.EQ.QE) GO TO 230
318.      IF(K2.EQ.QG .OR. K3.EQ.QG) GO TO 240
319.      IF(K2.EQ.QN .OR. K3.EQ.QN) GO TO 250
320.      GO TO 230
321.      220 XLB(NROW) = 0.
322.      XUB(NROW) = 1.E4
323.      GO TO 250
324.      230 XLB(NROW) = 0.
325.      XUB(NROW) = 0.
326.      GO TO 250
327.      240 XLB(NROW) = 0.
328.      XUB(NROW) = 1.E4
329.      A(NROW) = -1.
330.      GO TO 260
331.      250 A(NROW) = 1.
332.      260 IA(NROW) = NROW
333.      LA(NROW) = NROW
334.      JH(NROW) = NROW
335.      KINBAS(NROW) = NROW
336.      NELEM=NROW
337.      GO TO 5
338.      C
339.      C      MATRIX ELEMENTS
340.      C
341.      320 J = 3
342.      K = 4
343.      IF (DABS(ATEMP1) .LE. ZTOLZE) GO TO 321
344.      GO TO 324
345.      321 J=5
346.      K=6
347.      ATEMP1=ATEMP2
348.      GO TO 330
349.      324 CCNTINUE
350.      IF (NAME(1) .EQ. ICS1 .AND. NAME(2) .EQ. ICS2) GO TO 330
351.      C
352.      C      TEST FOR SPLIT VECTOR
353.      C
354.      DO 325 I = 1,NCOL
355.      IF (NAME(1) .NE. ICNAM(I,1)) GO TO 325
356.      IF (NAME(2) .NE. ICNAM(I,2)) GO TO 325
357.      WRITE(6,8250) NAME(1),NAME(2)
358.      8250 FORMAT(14H0SPLIT VECTOR ,2A4)
359.      325 CONTINUE
360.      NCOL = NCOL + 1
361.      ICS1 = NAME(1)
362.      ICS2 = NAME(2)
363.      ICNAM(NCOL,1) = ICS1
364.      ICNAM(NCOL,2) = ICS2
365.      LA(NCOL) = NELEM + 1

```



```

366. C
367. C      TEST FOR ROW MATCH
368. C
369. 330 DO 340 I = 1,NROW
370.     IF(NAME(J) .NE. ICNAM(I,1) .OR. NAME(K) .NE. ICNAM(I,2))GO TO 340
371.     NELEM = NELEM + 1
372.     IA(NELEM) = I
373.     A(NELEM) = ATEMP1
374.     LA(NCOL+1)=NELEM+1
375. 335 IF(K .GT. 5) GO TO 5
376.     IF(DABS(ATEMP2) .LE. ZTOLZE) GO TO 5
377.     J = 5
378.     K = 6
379.     ATEMP1 = ATEMP2
380.     GO TO 330
381. 340 CONTINUE
382.     WRITE(6,8300) NAME(J),NAME(K),NAME(1),NAME(2)
383. 8300 FORMAT(18HNO MATCH FOR ROW ,2A4,10HAT COLUMN ,2A4)
384.     STOP
385. C
386. C      BASIS CARDS
387. C
388. 410 DO 420 I = 1,NCOL
389.     IF(NAME(1) .NE. ICNAM(I,1) .OR. NAME(2) .NE. ICNAM(I,2)) GO TO 420
390.     IBVEC = I
391.     GO TO 425
392. 420 CONTINUE
393.     WRITE(6,8400) NAME(1),NAME(2)
394. 8400 FORMAT(21HNO MATCH FOR VECTOR ,2A4)
395.     GO TO 5
396. 425 DO 430 I = 1,NROW
397.     IF (NAME(3).NE.ICNAM(I,1).OR.NAME(4).NE.ICNAM(I,2)) GO TO 430
398.     IBROW=I
399.     GO TO 440
400. 430 CONTINUE
401.     WRITE(6,8300) NAME(3),NAME(4)
402.     GO TO 5
403. 440 JH(IBROW) = IBVEC
404.     KINBAS(IBROW) = 0
405.     KINBAS(IBVEC) = IBROW
406.     GO TO 5
407. C
408. C      RHS
409. C
410. 500 J = 3
411.     K = 4
412.     IF (DABS(ATEMP1) .LE. ZTOLZE) GO TO 521
413.     GO TO 524
414. 521 J=5
415.     K=6
416.     ATEMP1=ATEMP2
417.     GO TO 530
418. 524 CONTINUE
419.     IF (NAME(1) .EQ. ICS1 .AND. NAME(2) .EQ. ICS2) GO TO 530
420. C
421. C      TEST FOR SPLIT VECTOR
422. C
423. DO 525 I = 1,NCOL
424.     IF (NAME(1) .NE. ICNAM(I,1)) GO TO 525
425.     IF (NAME(2) .NE. ICNAM(I,2)) GO TO 525
426.     WRITE(6,8250) NAME(1),NAME(2)

```

```

427.      525 CONTINUE
428.      ICS1 = NAME(1)
429.      ICS2 = NAME(2)
430.      C
431.      C      TEST FOR ROW MATCH
432.      C
433.      530 DO 540 I = 1,NROW
434.      IF(NAME(J) .NE. ICNAM(I,1) .OR. NAME(K) .NE. ICNAM(I,2))GO TO 540
435.      B(I) = ATEMP1
436.      535 IF(K .GT. 5) GO TO 5
437.      IF(DABS(ATEMP2) .LE. ZTOLZE) GO TO 5
438.      J = 5
439.      K = 6
440.      ATEMP1 = ATEMP2
441.      GO TO 530
442.      540 CONTINUE
443.      WRITE(6,8300) NAME(J),NAME(K)
444.      STOP
445.      C
446.      C      END OF INPUT
447.      C
448.      600 NSCOL = NCOL - NROW
449.      K = NROW + 1
450.      C      INPUT LOWER AND UPPER BOUNDS ON DECISION VARIABLES FOR SUBPROB-
451.      C      LEM INPROB
452.      READ (5,650) (XLB(J), J=K,NCOL)
453.      READ (5,650) (XUB(J), J=K,NCOL)
454.      650 FORMAT (15F5.0)
455.      XLB(ICBJ) = -10000.
456.      XUB(ICBJ) = 10000.
457.      C      INPUT LOWER AND UPPER BOUNDS ON ALLOCATION VARIABLE FOR SUB-
458.      C      PROBLEM INPROB
459.      READ (5,651) IBLB(INPROB),IBUB(INPROB)
460.      651 FORMAT (8I10)
461.      TEMP = FLGAT(IBLB(INPROB))
462.      B(2) = DBLE(TEMP)
463.      INVFRQ = NROW
464.      C      STORE A,IA,LA,XUB,XLB,B,NCOL,NROW FOR SUBPROBLEM INPROB
465.      LL = JFIRST(INPROB) - 1
466.      DO 1010 I=1,NELEM
467.      LLPI = LL + I
468.      AMATRS(LLPI) = A(I)
469.      1010 IAS(LLPI) = IA(I)
470.      DO 1020 K=1,NCOL
471.      LAS(K,INPROB) = LA(K)
472.      XLBS(K,INPROB) = XLB(K)
473.      1020 XUBS(K,INPROB) = XUB(K)
474.      LL = NROW - 2
475.      DO 1030 I=1,LL
476.      1030 RHS(I,INPROB) = SNGL(B(I+2))
477.      NCOLS(INPROB) = NCOL
478.      NROWS(INPROB) = NROW
479.      JFIRST(INPROB+1) = JFIRST(INPROB) + NELEM
480.      C
481.      NELEM = NELEM - NROW
482.      RELEM = NELEM
483.      RDENS = KELEM/(NROW*NSCOL)
484.      RETURN
485.      END
486.      SUBROUTINE FTRAN(IPAR)
487.      C

```

```

488. C      PERFORMS FORWARD TRANSFORMATION ON COLUMN STORED IN VECTOR Y
489. C*****DESCRIPTION OF PARAMETERS*****
490. C      IPAR = PARAMETER INDICATING BY WHICH ETA-VECTORS THE MATRIX
491. C      COLUMN IS TO BE UPDATED
492. C*****
493. C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
494. C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
495. C
496. C      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
497. C      1      INTEGER*4 (I-N,Q)
498. C      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
499. C      DOUBLE PRECISION E(1000)
500. C      REAL A(500)
501. C
502. C      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSN,QRO,QMA,QBA,QFI,
503. C      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
504. C      2      QE,QH,QL,QD,QR,QM,QG,NTMAX
505. C
506. C      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
507. C      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
508. C      2      IOBJ,IRCWP,IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,ITSINV,IFFEZ,
509. C      3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
510. C      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
511. C
512. C      GO TO (100,110), IPAR
513. C      100 NFE = 1
514. C      NLE = NETA
515. C      GO TO 200
516. C      110 NFE = NLETA + 1
517. C      NLE = NETA
518. C      200 IF (NFE .GT. NLE) GO TO 9000
519. C      DO 1000 IK = NFE,NLE
520. C      LL = LE(IK)
521. C      KK = LE(IK+1) - 1
522. C      IPIV = IE(LL)
523. C      DY = Y(IPIV)
524. C      DY = DY/E(LL)
525. C      Y(IPIV) = DY
526. C      IF (KK .LE. LL) GO TO 1000
527. C      LL = LL + 1
528. C      DO 500 J = LL, KK
529. C      IR = IE(J)
530. C      Y(IR) = Y(IR) - E(J) * DY
531. C      500 CONTINUE
532. C      1000 CONTINUE
533. C      9000 CONTINUE
534. C      RETURN
535. C      END
536. C      SUBROUTINE BTRAN
537. C
538. C      PERFORMS BACKWARD TRANSFORMATION ON COLUMN STORED IN VECTOR Y
539. C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
540. C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
541. C
542. C      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
543. C      1      INTEGER*4 (I-N,Q)
544. C      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
545. C      DOUBLE PRECISION E(1000)
546. C      REAL A(500)
547. C
548. C      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSN,QRO,QMA,QBA,QFI,

```



```

549.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
550.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
551.  C
552.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
553.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
554.      2      IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,ITSINV,IFFEZ,
555.      3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
556.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
557.  C
558.      IF (NETA .LE. 0) GO TO 9000
559.      DO 1000 I = 1,NETA
560.      IK = NETA - I + 1
561.      LL = LE(IK)
562.      KK = LE(IK+1) - 1
563.      IPIV = IE(LL)
564.      DP = E(LL)
565.      DY = Y(IPIV)
566.      DSUM = 0.
567.      IF (KK .LE. LL) GO TO 600
568.      LL = LL + 1
569.      DO 500 J = LL,KK
570.      IR = IE(J)
571.      DE = E(J)
572.      DPROD = DE * Y(IR)
573.      DSUM = DSUM + DPROD
574.      500 CONTINUE
575.  C
576.      600 Y(IPIV) = (DY - DSUM) / DP
577.      1000 CONTINUE
578.      9000 RETURN
579.      END
580.      SUBROUTINE FORMC
581.  C
582.  C      FORMS COST VECTOR FOR CURRENT PRIMAL-SIMPLEX ITERATION; USED IN
583.  C      CONJUNCTION WITH SUBROUTINE PRICE
584.  C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
585.  C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
586.  C
587.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
588.      1      INTEGER*4 (I-N,Q)
589.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
590.      DOUBLE PRECISION E(1000)
591.      REAL A(500)
592.  C
593.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
594.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
595.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
596.  C
597.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
598.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
599.      2      IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,ITSINV,IFFEZ,
600.      3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
601.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
602.  C
603.      MSTAT = QI
604.      IFFEZ = 0
605.      SUMINF = 0.
606.      Y(IOBJ) = 0.
607.      DO 30 I=1,NROW
608.      IF (I .EQ. IOBJ) GO TO 30
609.      ICOL = JH(I)

```

```

610.      IF (X(I) .LE. (XLB(ICOL) - ZTOLZE)) GO TO 10
611.      IF (X(I) .GE. (XUB(ICOL) + ZTOLZE)) GO TO 20
612.      Y(I) = 0.
613.      GO TO 30
614. 10 Y(I) = 1.
615.      SUMINF = SUMINF + XLB(ICOL) - X(I)
616.      GO TO 30
617. 20 Y(I) = -1.
618.      SUMINF = SUMINF + X(I) - XUB(ICOL)
619. 30 CONTINUE
620.      IF (SUMINF .GT. ZTOLSM) RETURN
621.      Y(IOBJ) = 1.
622.      IFFEZ = 1
623.      MSTAT = QF
624.      RETURN
625.      END
626.      SUBROUTINE PRICE
627.  C
628.  C      PRICES GUT NONBASIC COLUMNS, CHOOSES PIVOT COLUMN JCOLP FOR
629.  C      CURRENT PRIMAL-SIMPLEX ITERATION
630.  C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
631.  C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
632.  C
633.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
634.      1      INTEGER*4 (I-N,Q)
635.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
636.      INTEGER*2 IPART,INCUMB,IVBND,IVID,IOBND
637.      DOUBLE PRECISION E(1000)
638.      REAL A(500)
639.  C
640.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
641.      1      QEO,CBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
642.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
643.  C
644.      COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
645.      1      NPIVOT,IPTYPE,RCOST,IFEAS,IPART(122),INCUMB(122),
646.      2      IVBND(500),IVID(500),IOBND(500)
647.  C
648.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
649.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
650.      2      IOBJ,IRCWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
651.      3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
652.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
653.  C
654.      CMIN = 1.E10
655.      CMAX = -1.E10
656.      DO 1000 J=1,NCOL
657.      IF (KINBAS(J) .GT. 0) GO TO 1000
658.      IF ((XUB(J) - XLB(J)) .LT. ZTOLZE) GO TO 1000
659.      DSUM = 0.
660.      LL = LA(J)
661.      KK = LA(J+1) - 1
662.      DO 500 I = LL,KK
663.      IR = IA(I)
664.      DE = A(I)
665.      DPROD = DE * Y(IR)
666.      DSUM = DSUM + DPROD
667. 500 CONTINUE
668.      IF (KINBAS(J) .EQ. -1) GO TO 600
669.      IF (DSUM .GE. CMIN) GO TO 1000
670.      CMIN = DSUM

```

```

671.      JCOL1 = J
672.      GO TO 1000
673.      600 IF (DSUM .LE. CMAX) GO TO 1000
674.      CMAX = DSUM
675.      JCOL2 = J
676.      1000 CONTINUE
677.      C
678.      IF (CMIN .LE. -ZTCOST) GO TO 1500
679.      IF (CMAX .GE. ZTCOST) GO TO 2000
680.      JCOLP = 0
681.      RCOST = 0.
682.      RETURN
683.      1500 IF (CMAX .GE. ZTCOST) GO TO 2500
684.      1600 JCOLP = JCOL1
685.      RCOST = CMIN
686.      RETURN
687.      2000 JCOLP = JCOL2
688.      RCOST = CMAX
689.      RETURN
690.      2500 IF (ABS(CMIN) - CMAX) 2000,2000,1600
691.      END
692.      SUBROUTINE CHUZR
693.      C
694.      C      PERFORMS MIN-RATIO TEST FOR PIVOT COLUMN JCOLP DETERMINED IN
695.      C      SUBROUTINE PRICE. SELECTS PIVOT ROW IROWP FOR CURRENT PRIMAL-
696.      C      SIMPLEX ITERATION.
697.      C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
698.      C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
699.      C
700.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
701.      1      INTEGER*4 (I-N,Q)
702.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
703.      INTEGER*2 IPART,INCUMB,IVBND,IVID,IOBND
704.      DOUBLE PRECISION E(1000)
705.      REAL A(500)
706.      C
707.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
708.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
709.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
710.      C
711.      COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
712.      1      NPIVOT,IPTYPE,RCOST,IFEAS,IPART(122),INCUMB(122),
713.      2      IVBND(500),IVID(500),IOBND(500)
714.      C
715.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
716.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
717.      2      IOBJ,IRCWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
718.      3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
719.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
720.      C
721.      IF (KINBAS(JCOLP) .EQ. -1) GO TO 1000
722.      C
723.      C      INCOMING VARIABLE AT LOWER BOUND
724.      C
725.      DP = 1.E10
726.      DO 500 I=1,NROW
727.      IF (I .EQ. IOBJ) GO TO 500
728.      ICOL = JH(I)
729.      IF (Y(I) .GT. ZTOLPV) GO TO 100
730.      IF (Y(I) .LT. -ZTOLPV) GO TO 200
731.      GO TO 500

```



```

732.      C          POSITIVE COEFFICIENT
733.      100 IF (X(I) .LT. (XLB(ICOL) - ZTOLZE)) GO TO 500
734.      DE = (X(I) - XLB(ICOL))/Y(I)
735.      IF (DE .GE. DP) GO TO 500
736.      IPTYPE = 0
737.      GO TO 250
738.      C          NEGATIVE COEFFICIENT
739.      200 IF (X(I) .GT. (XUB(ICOL) + ZTOLZE)) GO TO 500
740.      DE = (X(I) - XUB(ICOL))/Y(I)
741.      IF (DE .GE. DP) GO TO 500
742.      IPTYPE = -1
743.      250 DP = DE
744.      IROWP = I
745.      500 CONTINUE
746.      DE = CP + XLB(JCOLP)
747.      IF (DE .LT. XUB(JCOLP)) GO TO 600
748.      DP = XUB(JCOLP) - XLB(JCOLP)
749.      NPIVOT = 0
750.      RETURN
751.      600 NPIVOT = 1
752.      RETURN
753.      C
754.      C          INCOMING VARIABLE AT UPPER BOUND
755.      C
756.      1000 DP = -1.E10
757.      DO 1500 I=1,NROW
758.      IF (I .EQ. IOBJ) GO TO 1500
759.      ICOL = JH(I)
760.      IF (Y(I) .GT. ZTOLPV) GO TO 1100
761.      IF (Y(I) .LT. -ZTOLPV) GO TO 1200
762.      GO TO 1500
763.      C          POSITIVE COEFFICIENT
764.      1100 IF (X(I) .GT. (XUB(ICOL) + ZTOLZE)) GO TO 1500
765.      DE = (X(I) - XUB(ICOL))/Y(I)
766.      IF (DE .LE. DP) GO TO 1500
767.      IPTYPE = -1
768.      GO TO 1250
769.      C          NEGATIVE COEFFICIENT
770.      1200 IF (X(I) .LT. (XLB(ICOL) - ZTOLZE)) GO TO 1500
771.      DE = (X(I) - XLB(ICOL))/Y(I)
772.      IF (DE .LE. DP) GO TO 1500
773.      IPTYPE = 0
774.      1250 DP = DE
775.      IROWP = I
776.      1500 CONTINUE
777.      DE = CP + XUB(JCOLP)
778.      IF (DE .GT. XLB(JCOLP)) GO TO 1600
779.      DP = XLB(JCOLP) - XUB(JCOLP)
780.      NPIVOT = 0
781.      RETURN
782.      1600 NPIVOT = 1
783.      RETURN
784.      END
785.      SUBROUTINE WRETA
786.      C
787.      C          FORMS NEW ETA-VECTOR CORRESPONDING TO CURRENT SIMPLEX PIVOT
788.      C          SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
789.      C          BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
790.      C
791.      IMPLICIT REAL*4 (A,C,E-H,G,P,R-W,Z), REAL*8 (B,D,X,Y),
792.      I          INTEGER*4 (I-N,Q)

```

```

793.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
794.      DOUBLE PRECISION E(1000)
795.      REAL A(500)
796.
797.      C      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
798.      1      QEO,CBL,QA,QPL,QMI,CZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
799.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
800.
801.      C      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
802.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
803.      2      IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
804.      3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
805.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
806.
807.      C      NELEM = NELEM + 1
808.      IE(NELEM) = IROWP
809.      E(NELEM) = Y(IROWP)
810.
811.      C      DO 1000 I = 1,NROW
812.      IF (I .EQ. IROWP) GO TO 1000
813.      IF (DABS(Y(I)) .LE. ZTOLZE) GO TO 1000
814.      NELEM = NELEM + 1
815.      IE(NELEM) = I
816.      E(NELEM) = Y(I)
817.      1000 CONTINUE
818.
819.      C      NETA = NETA + 1
820.      LE(NETA+1) = NELEM + 1
821.      RETURN
822.      END
823.      SUBROUTINE SHIFTR(IOLD,INEW)
824.
825.      C      REARRANGES DATA STORAGE
826.      C*****DESCRIPTION OF PARAMETERS*****
827.      C      IOLD, INEW = PARAMETERS INDICATING STORAGE LOCATIONS FROM AND
828.      C      TO WHICH DATA IS TO BE TRANSFERRED, RESPECTIVELY
829.      C*****
830.      C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
831.      C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
832.
833.      C      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
834.      1      INTEGER*4 (I-N,Q)
835.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
836.      DOUBLE PRECISION E(1000)
837.      REAL A(500)
838.
839.      C      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
840.      1      QEO,CBL,QA,QPL,QMI,CZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
841.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
842.
843.      C      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
844.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
845.      2      IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
846.      3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
847.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
848.
849.      C      DIMENSION BARRAY(240)
850.      EQUIVALENCE (BARRAY(1),B(1))
851.      IFD = (IOLD - 1) * NRMAX
852.      IFN = (INEW - 1) * NRMAX
853.
854.      C

```

```

854.      DO 1000 I = 1,NROW
855.      BARRAY(IFN + I) = BARRAY(IFD + I)
856.      1000 CONTINUE
857.      RETURN
858.      END
859.      SUBROUTINE INVERT
860.      C
861.      C      COMPUTES INVERSE OF CURRENT BASIS BY LU DECOMPOSITION
862.      C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
863.      C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
864.      C
865.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
866.      1      INTEGER*4 (I-N,Q)
867.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
868.      DOUBLE PRECISION E(1000)
869.      REAL A(500)
870.      C
871.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRD,QMA,QBA,QFI,
872.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
873.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
874.      C
875.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
876.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
877.      2      IOBJ,IRCWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
878.      3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
879.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
880.      C
881.      INTEGER*2 MREG,HREG,VREG
882.      DIMENSION MREG(60),HREG(60),VREG(60)
883.      EQUIVALENCE (MREG(1),YTEMP(1)),(HREG(1),YTEMP(31)),(VREG(1),X(1))
884.      C
885.      C      SET PARAMETERS
886.      C
887.      NETA = 0
888.      NLETA = 0
889.      NUETA = 0
890.      NELEM = 0
891.      NLELEM = 0
892.      NUELEM = 0
893.      NABOVE = 0
894.      LE(1) = 1
895.      LR1 = 1
896.      KR1 = 0
897.      LR4 = NROW + 1
898.      KR4 = NROW
899.      C
900.      C      PUT SLACKS AND ARTIFICIALS IN PART 4 AND REST IN PART 1
901.      C
902.      DO 100 I = 1,NROW
903.      IF (JH(I) .GT. NROW) GO TO 50
904.      LR4 = LR4 - 1
905.      MREG(LR4) = JH(I)
906.      VREG(LR4) = JH(I)
907.      GO TO 90
908.      50 KR1 = KR1 + 1
909.      VREG(KR1) = JH(I)
910.      90 HREG(I) = -1
911.      JH(I) = 0
912.      100 CONTINUE
913.      C
914.      KR3 = LR4 - 1

```



```

915.      LR3 = LR4
916.      C
917.      DO 200 I = LR4,KR4
918.      IR = MREG(I)
919.      HREG(IR) = 0
920.      JH(IR) = IR
921.      KINBAS(IR) = IR
922.      200 CONTINUE
923.      C
924.      C      PULL OUT VECTORS BELOW BUMP AND GET ROW COUNTS
925.      C
926.      NBNONZ = KR4 - LR4 + 1
927.      IF (KR1 .EQ. 0) GO TO 1190
928.      J = LR1
929.      210 IV = VREG(J)
930.      LL = LA(IV)
931.      KK = LA(IV+1) - 1
932.      IRCNT = 0
933.      DO 220 I = LL, KK
934.      NBNONZ = NBNONZ + 1
935.      IR = IA(I)
936.      IF (HREG(IR) .GE. 0) GO TO 220
937.      IRCNT = IRCNT + 1
938.      HREG(IR) = HREG(IR) - 1
939.      IRP = IR
940.      220 CONTINUE
941.      IF (IRCNT - 1) 230,250,300
942.      230 WRITE(6,8000)
943.      8000 FORMAT(16HOMATRIX SINGULAR )
944.      KINBAS(IV) = 0
945.      VREG(J) = VREG(KR1)
946.      KR1 = KR1 - 1
947.      IF (J .GT. KR1) GO TO 310
948.      GO TO 210
949.      C
950.      250 VREG(J) = VREG(KR1)
951.      KR1 = KR1 - 1
952.      LR3 = LR3 - 1
953.      VREG(LR3) = IV
954.      MREG(LR3) = IRP
955.      HREG(IRP) = 0
956.      JH(IRP) = IV
957.      KINBAS(IV) = IRP
958.      IF (J .GT. KR1) GO TO 310
959.      GO TO 210
960.      300 IF (J .GE. KR1) GO TO 310
961.      J = J+1
962.      GO TO 210
963.      C
964.      C      PULL OUT REMAINING VECTORS ABOVE AND BELOW THE
965.      C      BUMP AND ESTABLISH MERIT COUNTS OF COLUMNS
966.      C
967.      310 NVREM = 0
968.      IF (KR1 .EQ. 0) GO TO 1190
969.      J = LR1
970.      320 IV = VREG(J)
971.      LL = LA(IV)
972.      KK = LA(IV+1) - 1
973.      IRCNT = 0
974.      DO 800 I = LL, KK
975.      IR = IA(I)

```

```

976.      IF(HREG(IR) .NE. -2) GO TO 400
977.      C
978.      C          PIVOT ABOVE BUMP (PART OF L)
979.      C
980.      NABOVE = NABOVE + 1
981.      IROWP = IR
982.      CALL LNPACK(IV)
983.      CALL WRETA
984.      NLETA = NETA
985.      JH(IR) = IV
986.      KINBAS(IV) = IR
987.      VREG(J) = VREG(KR1)
988.      KR1 = KR1 - 1
989.      NVREM = NVREM + 1
990.      HREG(IR) = IV
991.      GO TO 940
992.      C
993.      400 IF (HREG(IR) .GE. 0) GO TO 800
994.      IRCNT = IRCNT + 1
995.      IRP = IR
996.      800 CONTINUE
997.      C
998.      IF (IRCNT - 1) 810,900,1000
999.      810 WRITE(6,8000)
1000.      KINBAS(IV) = 0
1001.      VREG(J) = VREG(KR1)
1002.      NVREM = NVREM + 1
1003.      KR1 = KR1 - 1
1004.      IF (J .GT. KR1) GO TO 1010
1005.      GO TO 320
1006.      C
1007.      C          PUT VECTOR BELOW BUMP
1008.      C
1009.      900 VREG(J) = VREG(KR1)
1010.      NVREM = NVREM + 1
1011.      KR1 = KR1 - 1
1012.      LR3 = LR3 - 1
1013.      VREG(LR3) = IV
1014.      HREG(LR3) = IRP
1015.      HREG(IRP) = 0
1016.      JH(IRP) = IV
1017.      KINBAS(IV) = IRP
1018.      C
1019.      C          CHANGE ROW COUNTS
1020.      C
1021.      940 DO 950 II = LL, KK
1022.      IIR = IA(II)
1023.      IF (HREG(IIR) .GE. 0) GO TO 950
1024.      HREG(IIR) = HREG(IIR) + 1
1025.      950 CONTINUE
1026.      IF (J .GT. KR1) GO TO 1010
1027.      GO TO 320
1028.      1000 IF (J .GE. KR1) GO TO 1010
1029.      J = J+1
1030.      GO TO 320
1031.      1010 IF(NVREM .GT. 0) GO TO 310
1032.      C
1033.      C          GET MERIT COUNTS
1034.      C
1035.      1020 IF (KR1 .EQ. 0) GO TO 1190
1036.      DO 1100 J = LR1, KR1

```

```

1037.      IV = VREG(J)
1038.      LL = LA(IV)
1039.      KK = LA(IV+1) - 1
1040.      IMCNT = 0
1041.      DO 1050 I = LL, KK
1042.      IR = IA(I)
1043.      IF (HREG(IR) .GE. 0) GO TO 1050
1044.      IMCNT = IMCNT - (HREG(IR) + 1)
1045. 1050 CONTINUE
1046.      MREG(J) = IMCNT
1047. 1100 CONTINUE
1048. C
1049. C      SORT COLUMNS INTO MERIT ORDER
1050. C      USING SHELL SORT
1051. C
1052.      ISD = 1
1053. 1106 IF (KR1 .LT. 2*ISD) GO TO 1108
1054.      ISD = 2*ISD
1055.      GO TO 1106
1056. 1108 ISD = ISD - 1
1057. C      END OF INITIALIZATION
1058. 1101 IF (ISD .LE. 0) GO TO 1107
1059.      ISK = 1
1060. 1102 ISJ = ISK
1061.      ISL = ISK + ISD
1062.      ISY = MREG(ISL)
1063.      ISZ = VREG(ISL)
1064. 1103 IF (ISY .LT. MREG(ISJ)) GO TO 1104
1065. 1105 ISL = ISJ + ISD
1066.      MREG(ISL) = ISY
1067.      VREG(ISL) = ISZ
1068.      ISK = ISK + 1
1069.      IF ((ISK + ISD) .LE. KR1) GO TO 1102
1070.      ISD = (ISD - 1) / 2
1071.      GO TO 1101
1072. 1104 ISL = ISJ + ISD
1073.      MREG(ISL) = MREG(ISJ)
1074.      VREG(ISL) = VREG(ISJ)
1075.      ISJ = ISJ - ISD
1076.      IF (ISJ .GT. 0) GO TO 1103
1077.      GO TO 1105
1078. 1107 CONTINUE
1079. C      END OF SORT ROUTINE
1080. C
1081. C      PUT OUT BELOW BUMP ETAS (PART OF U)
1082. C
1083. 1150 NSLCK = 0
1084.      NBELOW = 0
1085.      NELAST = NEMAX
1086.      NTLAST = NTMAX
1087.      LE(NTLAST + 1) = NELAST + 1
1088. C
1089.      LR = LR3
1090.      IF (LR3 .GE. LR4) LR = LR4
1091.      IF (LR .GT. KR4) GO TO 2050
1092.      JK = KR4 + 1
1093.      DO 2000 JJ = LR, KR4
1094.      JK = JK - 1
1095.      IV = VREG(JK)
1096.      I = MREG(JK)
1097.      NBELOW = NBELOW + 1

```



```

1098.      IF (IV .GT. NROW) GO TO 1200
1099.      NSLCK = NSLCK + 1
1100.      1200 LL = LA(IV)
1101.      KK = LA(IV+1) - 1
1102.      IF (KK .GT. LL) GO TO 1300
1103.      1250 IF (ABS(A(LL) - 1.) .LE. ZTOLZE) GO TO 2000
1104.      C
1105.      1300 NUETA = NUETA + 1
1106.      DO 1400 J = LL, KK
1107.      IR = IA(J)
1108.      IF (IR .EQ. 1) GO TO 1390
1109.      IE(NELAST) = IR
1110.      E(NELAST) = A(J)
1111.      NELAST = NELAST + 1
1112.      NUELEM = NUELEM + 1
1113.      GO TO 1400
1114.      1390 EP = A(J)
1115.      1400 CONTINUE
1116.      IE(NELAST) = I
1117.      E(NELAST) = EP
1118.      LE(NLAST) = NELAST
1119.      NELAST = NELAST + 1
1120.      NTLAST = NTLAST + 1
1121.      NUELEM = NUELEM + 1
1122.      2000 CONTINUE
1123.      2050 IF (KRI .EQ. 0) GO TO 3500
1124.      C
1125.      C      DC L-U DECOMPOSITION OF BUMP
1126.      C
1127.      DO 3000 J = LRI, KRI
1128.      IV = VREG(J)
1129.      CALL UNPACK(IV)
1130.      CALL FTRAN(2)
1131.      IROWP = 0
1132.      IRCMIN = -999999
1133.      DO 2100 I = 1, NROW
1134.      IF (DABS(Y(I)) .LE. ZTOLPV) GO TO 2100
1135.      IF (HREG(I) .GE. 0) GO TO 2100
1136.      IF (HREG(I) .LE. IRCMIN) GO TO 2100
1137.      IRCMIN = HREG(I)
1138.      IROWP = I
1139.      2100 CONTINUE
1140.      IF (IROWP .GT. 0) GO TO 2150
1141.      WRITE(6, 8000)
1142.      KINBAS(IV) = 0
1143.      GO TO 3000
1144.      C
1145.      2150 INCR = HREG(IROWP) + 3
1146.      C
1147.      C      WRITE L AND U ETAS
1148.      C
1149.      IF (J .EQ. KRI) GO TO 2160
1150.      NELEM = NELEM + 1
1151.      IE(NELEM) = IROWP
1152.      E(NELEM) = Y(IROWP)
1153.      2160 DO 2300 I = 1, NROW
1154.      IF (I .EQ. IROWP) GO TO 2300
1155.      IF (DABS(Y(I)) .LE. ZTOLZE) GO TO 2300
1156.      IF (HREG(I) .GE. 0) GO TO 2200
1157.      C
1158.      C      L ETA ELEMENTS

```

```

1159. C
1160.      NELEM = NELEM + 1
1161.      IE(NELEM) = I
1162.      E(NELEM) = Y(I)
1163.      GO TO 2300
1164. C
1165. C          U ETA ELEMENTS
1166. C
1167.      2200 IE(NELAST) = I
1168.      E(NELAST) = Y(I)
1169.      NELAST = NELAST - 1
1170.      NUELEM = NUELEM + 1
1171.      2300 CONTINUE
1172. C
1173.      JH(IROWP) = IV
1174.      KINBAS(IV) = IROWP
1175.      NUETA = NUETA + 1
1176.      IE(NELAST) = IROWP
1177.      IF (J.NE. KR1) GO TO 2330
1178.      E(NELAST) = Y(IROWP)
1179.      GO TO 2340
1180.      2330 E(NELAST) = 1.
1181.      NETA = NETA + 1
1182.      LE(NETA+1) = NELEM + 1
1183.      2340 NUELEM = NUELEM + 1
1184.      LE(NTLAST) = NELAST
1185.      NELAST = NELAST - 1
1186.      NTLAST = NTLAST - 1
1187. C
1188. C          UPDATE ROW COUNTS
1189. C
1190.      DO 2350 I = 1,NROW
1191.      IF (DABS(Y(I)) .LE. 2TOLZE) GO TO 2350
1192.      IF (HREG(I) .GE. 0) GO TO 2350
1193.      HREG(I) = HREG(I) - INCR
1194.      IF (HREG(I) .GE. 0) HREG(I) = -1
1195.      2350 CONTINUE
1196.      HREG(IROWP) = 0
1197.      3000 CONTINUE
1198. C
1199. C          MERGE L AND U ETAS
1200. C
1201.      3500 NLETA = NETA
1202.      NETA = NLETA + NUETA
1203.      NLELEM = NELEM
1204.      NELEM = NLELEM + NUELEM
1205.      IF (NUELEM .EQ. 0) GO TO 3550
1206.      CALL SHFTE
1207. C
1208. C          INSERT SLACKS FOR DELETED COLUMNS
1209. C
1210.      3550 DO 3600 I = 1,NROW
1211.      IF (JH(I) .NE. 0) GO TO 3600
1212.      JH(I) = I
1213.      IROWP = I
1214.      CALL UNPACK(I)
1215.      CALL FTRAN(1)
1216.      CALL WRETA
1217.      3600 CONTINUE
1218. C
1219. C          UPDATE X

```

```

1220. C
1221. CALL SHIFTR(1,3)
1222. DO 9000 J=1,NCOL
1223. IF (KINBAS(J)) 8600,8700,9000
1224. 8600 DE = XUB(J)
1225. GO TO 8750
1226. 8700 DE = XLB(J)
1227. 8750 LL = LA(J)
1228. KK = LA(J+1) - 1
1229. DO 8800 I=LL, KK
1230. IR = IA(I)
1231. 8800 Y(IR) = Y(IR) - A(I)*DE
1232. 9000 CONTINUE
1233. CALL FTRAN(1)
1234. CALL SHIFTR(3,2)
1235. RETURN
1236. END
1237. SUBROUTINE UNPACK(IV)
1238. C
1239. C EXPANDS COMPRESSED MATRIX COLUMN
1240. C*****DESCRIPTION OF PARAMETERS*****
1241. C IV = PARAMETER INDEXING COLUMN TO BE EXPANDED
1242. C*****
1243. C SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
1244. C BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1245. C
1246. C IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1247. C 1 INTEGER*4 (I-N,Q)
1248. C INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1249. C DOUBLE PRECISION E(1000)
1250. C REAL A(500)
1251. C
1252. C COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1253. C 1 GEO,CBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1254. C 2 GE,QH,QL,QO,QR,QM,QG,NTMAX
1255. C
1256. C COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
1257. C 1 DY,DE,DP,A,IGNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1258. C 2 IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,ITSINV,IFFEZ,
1259. C 3 JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1260. C 4 IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1261. C
1262. C DO 100 I = 1,NROW
1263. C Y(I) = 0.
1264. C 100 CONTINUE
1265. C
1266. C LL = LA(IV)
1267. C KK = LA(IV+1) - 1
1268. C DO 200 I = LL, KK
1269. C IR = IA(I)
1270. C Y(IR) = A(I)
1271. C 200 CONTINUE
1272. C
1273. C RETURN
1274. C END
1275. C SUBROUTINE SHFTE
1276. C
1277. C SUBROUTINE FOR INVERT
1278. C SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
1279. C BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1280. C

```



```

1281.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1282.      1      INTEGER*4 (I-N,Q)
1283.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1284.      DOUBLE PRECISION E(1000)
1285.      REAL A(500)
1286.      C
1287.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1288.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1289.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
1290.      C
1291.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
1292.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1293.      2      IOBJ,IRCOMP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
1294.      3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1295.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1296.      C
1297.      C      SHIFT IE AND E OF U ELEMENTS
1298.      C
1299.      NF = NEMAX - NUELEM + 1
1300.      INCR = 0
1301.      DO 1000 I = NF,NEMAX
1302.      INCR = INCR + 1
1303.      IE(NLELEM + INCR) = IE(I)
1304.      E(NLELEM + INCR) = E(I)
1305.      1000 CONTINUE
1306.      C
1307.      IDIF = NEMAX - NLELEM - NUELEM
1308.      NF = NTMAX - NUETA + 1
1309.      INCR = 0
1310.      DO 2000 I = NF,NTMAX
1311.      INCR = INCR + 1
1312.      LE(NLETA + INCR) = LE(I) - IDIF
1313.      2000 CONTINUE
1314.      LE(NETA+1) = NELEM + 1
1315.      RETURN
1316.      END
1317.      SUBROUTINE UPBETA
1318.      C
1319.      C      UPDATES RIGHT-HAND SIDES TO REFLECT NEW BASIS RESULTING FROM
1320.      C      CURRENT SIMPLEX PIVOT
1321.      C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
1322.      C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1323.      C
1324.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1325.      1      INTEGER*4 (I-N,Q)
1326.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1327.      INTEGER*2 IPART,INCUMB,IVBND,IVIO,ICBND
1328.      DOUBLE PRECISION E(1000)
1329.      REAL A(500)
1330.      C
1331.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1332.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1333.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
1334.      C
1335.      COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1336.      1      NPIVOT,IPTYPE,RCOST,IFEAS,IPART(122),INCUMB(122),
1337.      2      IVBND(500),IVIO(500),IOBND(500)
1338.      C
1339.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
1340.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1341.      2      IOBJ,IRCOMP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,

```

```

1342.      3      JCCLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1343.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1344.  C
1345.      DC 1000 I=1,NROW
1346.  1000 X(I) = X(I) - Y(I)*DP
1347.      IF (NPIVOT .EQ. 1) GO TO 2000
1348.      KINBAS(JCOLP) = -(KINBAS(JCOLP) + 1)
1349.      IVOUT = JCOLP
1350.      RETURN
1351.  2000 X(IROWP) = DE
1352.      IVOUT = JH(IROWP)
1353.      KINBAS(JCOLP) = IRCWP
1354.      KINBAS(IVOUT) = IPTYPE
1355.      JH(IRCWP) = JCOLP
1356.      RETURN
1357.      END
1358.      SUBROUTINE NORMAL
1359.  C
1360.  C      SERVES AS MASTER PROGRAM FOR LINEAR PROGRAMMING COMPONENT
1361.  C      (REVISED PRIMAL-SIMPLEX METHOD).
1362.  C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
1363.  C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1364.  C
1365.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1366.  1      INTEGER*4 (I-N,Q)
1367.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1368.      INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
1369.      DOUBLE PRECISION E(1000)
1370.      REAL A(500)
1371.  C
1372.      COMMON/BLOCK1/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTCLSM,QRO,QMA,QBA,QFI,
1373.  1      GEO,CBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1374.  2      QE,QH,QL,QO,QR,QM,QG,NTMAX
1375.  C
1376.      COMMON/BLOCK2/ DEPART(122),REV8ND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1377.  1      NPIVOT,IPTYPE,RCOST,IFEAS,IPART(122),INCUMB(122),
1378.  2      IVBND(500),IVID(500),IOBND(500)
1379.  C
1380.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
1381.  1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1382.  2      IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
1383.  3      JCCLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1384.  4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1385.  C
1386.      IF (ITSINV .LT. INVFRQ) GO TO 1500
1387.  1000 CALL INVERT
1388.      ITSINV = 0
1389.  C
1390.  C      SIMPLEX CYCLE
1391.  C
1392.  1500 CALL FORMC
1393.      CALL BTRAN
1394.      CALL PRICE
1395.      IF (JCCLP .GT. 0) GO TO 3000
1396.      IF (MSTAT .EQ. QI) GO TO 2000
1397.      MSTAT = QBL
1398.      GO TO 6000
1399.  2000 MSTAT = QN
1400.      GO TO 6000
1401.  3000 IVIN = JCCLP
1402.      CALL UNPACK(JCCLP)

```

```

1403.      CALL FTRAN(1)
1404.      CALL CHUZR
1405.      CALL UPBETA
1406.      ITCNT = ITCNT + 1
1407.      ITSINV = ITSINV + 1
1408.      IF (NPIVOT .EQ. 0) GO TO 4010
1409.      IF (NELEM .GT. 5000) GO TO 1000
1410.      CALL WRETA
1411.      4010 IF (ITSINV .GE. INVRFRQ) GO TO 1000
1412.      IF (ITCNT .GE. ITRFRQ) GO TO 6000
1413.      GO TO 1500
1414.
1415.      C
1416.      6000 RETURN
1417.      END
1418.      SUBROUTINE BANDB(INITBD)
1419.      C
1420.      C      MASTER PROGRAM FOR BRANCH-AND-BOUND INTEGER PROGRAMMING ROUTINE
1421.      C      ALSO SERVES AS MASTER PROGRAM FOR REOPTIMIZATION VIA DUAL-
1422.      C      SIMPLEX METHOD AFTER A FORWARD BRANCH.
1423.      C*****DESCRIPTION OF PARAMETERS*****
1424.      C      INITBD = INITIAL LOWER BOUND ON MAXIMAL OBJECTIVE VALUE (INPUT)
1425.      C*****
1426.      C
1427.      IMPLICIT REAL*4 (A,C,E-H,G,P,R-W,Z), REAL*8 (B,D,X,Y),
1428.      1      INTEGER*4 (I-N,Q)
1429.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1430.      INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
1431.      DOUBLE PRECISION E(1000)
1432.      REAL A(500)
1433.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1434.      1      QEO,CBL,QA,QPL,QM1,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1435.      2      QE,QH,QL,QO,QK,QM,QG,NTMAX
1436.      COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1437.      1      NPIVOT,IPTYPE,RCOST,IFEAS,IPART(122),INCUMB(122),
1438.      2      IVBND(500),IVID(500),IOBND(500)
1439.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
1440.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1441.      2      IOBJ,IRGWP,IVIN,IVOUT,ITCNT,INVRFRQ,ITRFRQ,ITSINV,IFFEZ,
1442.      3      JCPLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1443.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1444.      LISTL = 0
1445.      INCVAL = INITBD
1446.      C
1447.      C      TEST FOR FATHOMING
1448.      100 CALL TESTX
1449.      IF (MSTAT .EQ. QBL) GO TO 200
1450.      C
1451.      C      CURRENT NODE FATHOMED; BACKTRACK TO LAST PROMISING NODE ON LIST
1452.      150 CALL BKTRAK
1453.      IF LIST IS EMPTY, RETURN TO MAIN (COMPUTATIONS COMPLETED)
1454.      C
1455.      IF (LISTL .EQ. 0) RETURN
1456.      C
1457.      C      USE PRIMAL-SIMPLEX METHOD FOR REOPTIMIZATION AT NEW NODE
1458.      CALL NORMAL
1459.      GO TO 100
1460.      C
1461.      C      CURRENT NODE NOT FATHOMED; COMPUTE PENALTIES
1462.      C      BRANCHING AT CURRENT NODE IS DONE FROM SUBROUTINE PENLTS.
1463.      200 CALL PENLTS
1464.      IF (ICIR) 400,150,400
1465.      C
1466.      C      REINVERT CURRENT BASIS
1467.      1000 CALL INVERT
1468.      ITSINV = 0
1469.      C

```



```

1464. C          DUAL SIMPLEX CYCLE
1465. C
1466. C          CHOOSE PIVOT ROW IROWP
1467. 3CG CALL DCHUZR
1468.      IF (IROWP .GT. 0) GO TO 400
1469.      MSTAT = QBL
1470.      GO TO 100
1471. C          CHOOSE PIVOT COLUMN JCOLP
1472. 4CG CALL DCHUZR
1473.      IF (JCOLP .EQ. 0) GO TO 150
1474.      IVIN = JCOLP
1475. C          UPDATE RIGHT-HAND SIDES TO REFLECT NEW BASIS RESULTING FROM
1476. C          CURRENT SIMPLEX PIVOT
1477.      CALL UPBETA
1478.      ITCNT = ITCNT + 1
1479.      ITSINV = ITSINV + 1
1480. C          DECIDE WHETHER TO REINVERT CURRENT BASIS
1481.      IF ((NELEM .GT. 5600) .OR. (ITSINV .GE. INVFRQ)) GO TO 1000
1482. C          REINVERSION NOT NECESSARY YET; WRITE OUT NEW ETA-VECTOR FOR
1483. C          CURRENT SIMPLEX PIVOT
1484.      CALL WRETA
1485.      GO TO 300
1486.      END
1487.      SUBROUTINE DCHUZR
1488. C
1489. C          SELECTS PIVOT ROW IROWP FOR CURRENT DUAL-SIMPLEX ITERATION.
1490. C          SETS IROWP = 0 IF CURRENT BASIS IS OPTIMAL. OTHERWISE, IROWP
1491. C          IS CHOSEN TO BE THE ROW WITH GREATEST PRIMAL-INFEASIBILITY
1492. C
1493.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1494.      1      INTEGER*4 (I-N,Q)
1495.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1496.      INTEGER*2 IPART,INCUMB,IVBND,IVID,IOBND
1497.      DOUBLE PRECISION E(1000)
1498.      REAL A(500)
1499.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTCLSM,QRQ,QMA,QBA,QF1,
1500.      1      CEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1501.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
1502.      COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1503.      1      NPivot,IPTYPE,RCCST,IFEAS,IPART(122),INCUMB(122),
1504.      2      IVBND(500),IVID(500),IOBND(500)
1505.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
1506.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1507.      2      IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
1508.      3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1509.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1510. C
1511. C          CHOOSE ROW WITH GREATEST INFEASIBILITY
1512. C
1513.      IROWP = 0
1514.      DP = -1.E10
1515.      DO 1000 I=1,NROW
1516.      IF (I .EQ. IOBJ) GO TO 1000
1517.      ICOL = JH(I)
1518.      IF (X(I) .LT. (XLB(ICOL) - ZTOLZE)) GO TO 100
1519.      IF (X(I) .GT. (XUB(ICOL) + ZTOLZE)) GO TO 200
1520.      GO TO 1000
1521. C
1522. C          BASIC VARIABLE ON ROW I FALLS BELOW ITS LOWER BOUND
1523. 100 DE = XLB(ICOL) - X(I)
1524.      IF (DE .LE. DP) GO TO 1000

```

```

1525.          IPTYPE = 0
1526.          GO TO 250
1527.      C
1528.      C          BASIC VARIABLE ON ROW I EXCEEDS ITS UPPER BOUND
1529.      200 DE = X(I) - XUB(ICOL)
1530.          IF (DE .LE. DP) GO TO 1000
1531.          IPTYPE = -1
1532.      C
1533.          250 IROWP = I
1534.      1000 CONTINUE
1535.          RETURN
1536.      END
1537.      SUBROUTINE DCHU2C
1538.      C
1539.      C          SELECTS PIVOT COLUMN JCOLD FOR CURRENT DUAL-SIMPLEX ITERATION.
1540.      C          SETS JCOLD = 0 IF LP-PROBLEM AT CURRENT NODE IS INFEASIBLE.
1541.      C          OTHERWISE, CHOOSES JCOLD TO MAINTAIN PRIMAL-OPTIMALITY
1542.      C
1543.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1544.      1      INTEGER*4 (I-N,Q)
1545.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1546.      INTEGER*2 IPART,INCUMB,IVBND,IVID,IOBND
1547.      DOUBLE PRECISION E(1000)
1548.      REAL A(500)
1549.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1550.      1      QEO,QBL,QA,QPL,QM1,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1551.      2      QE,QH,QL,QO,QK,QM,QG,NTMAX
1552.      COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1553.      1      NPIVOT,IPTYPE,KCOST,IFEAS,IPART(122),INCUMB(122),
1554.      2      IVBND(500),IVID(500),IOBND(500)
1555.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
1556.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1557.      2      IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,ITSINV,IFFEZ,
1558.      3      JCOLD,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1559.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1560.      C
1561.      JCOLD = 0
1562.      IF (IPTYPE .EQ. -1) GO TO 1000
1563.      C
1564.      C          LEAVING VARIABLE FALLS BELOW ITS LOWER BOUND
1565.      C
1566.      DP = -1.E10
1567.      DO 500 J=1,NCOL
1568.      IF (KINBAS(J) .GT. 0) GO TO 500
1569.      IF ((XUB(J) - XLB(J)) .LE. ZTOLZE) GO TO 500
1570.      CALL UNPACK(J)
1571.      CALL FTRAN(1)
1572.      IF (KINBAS(J) .EQ. -1) GO TO 200
1573.      IF (Y(IROWP) + ZTOLPV) 225,225,500
1574.      200 IF (Y(IROWP) - ZTOLPV) 500,225,225
1575.      C
1576.      225 DE = Y(IOBJ)/Y(IROWP)
1577.      IF (DE - DP) 500,500,250
1578.      250 JCOLD = J
1579.      DP = DE
1580.      500 CONTINUE
1581.      C
1582.      IF (JCOLD .EQ. 0) RETURN
1583.      CALL UNPACK(JCOLD)
1584.      CALL FTRAN(1)
1585.      ICOL = JH(IROWP)

```

```

1586.      DP = (X(IROWP) - XLB(ICOL))/Y(IROWP)
1587.      GO TO 2000
1588.      C
1589.      C          LEAVING VARIABLE EXCEEDS ITS UPPER BOUND
1590.      C
1591.      1000 DP = 1.E10
1592.      DO 1500 J=1,NCOL
1593.      IF (KINBAS(J) .GT. 0) GO TO 1500
1594.      IF ((XUB(J) - XLB(J)) .LE. ZTOLZE) GO TO 1500
1595.      CALL UNPACK(J)
1596.      CALL FTRAN(1)
1597.      IF (KINBAS(J) .EQ. -1) GO TO 1200
1598.      IF (Y(IROWP) - ZTOLPV) 1500,1225,1225
1599.      1200 IF (Y(IROWP) + ZTOLPV) 1225,1225,1500
1600.      C
1601.      1225 DE = Y(IOBJ)/Y(IROWP)
1602.      IF (DE - DP) 1250,1500,1500
1603.      1250 JCOLP = J
1604.      DP = DE
1605.      1500 CONTINUE
1606.      C
1607.      IF (JCOLP .EQ. 0) RETURN
1608.      CALL UNPACK(JCOLP)
1609.      CALL FTRAN(1)
1610.      ICOL = JH(IROWP)
1611.      DP = (X(IROWP) - XUB(ICOL))/Y(IROWP)
1612.      C
1613.      2000 IF (KINBAS(JCOLP) .EQ. 0) DE = DP + XLB(JCOLP)
1614.      IF (KINBAS(JCOLP) .EQ. -1) DE = DP + XUB(JCOLP)
1615.      NPIVOT = 1
1616.      RETURN
1617.      END
1618.      SUBROUTINE TESTX
1619.      C
1620.      C          TESTS LP-OPTIMAL SOLUTION AT CURRENT NODE FOR FATHOMING.
1621.      C          FATHOMING OCCURS IF
1622.      C          (1) LP PROBLEM AT CURRENT NODE IS INFEASIBLE (MSTAT = QN); OR
1623.      C          (2) LP-OPTIMAL OBJECTIVE VALUE (IVAL) .LE. OBJECTIVE VALUE OF
1624.      C          CURRENT INCUMBENT SOLUTION (INCVAL); OR
1625.      C          (3) LP-OPTIMAL SOLUTION SATISFIES INTEGER RESTRICTIONS.
1626.      C
1627.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1628.      1      INTEGER*4 (I-N,Q)
1629.      INTEGER*2 JH,KINBAS,IA,LE,IA,IE
1630.      INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
1631.      DOUBLE PRECISION E(1000)
1632.      REAL A(500)
1633.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCGST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1634.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1635.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
1636.      COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1637.      1      NPIVOT,IPTYPE,RCOST,IFEAS,IPART(122),INCUMB(122),
1638.      2      IVBND(500),IVID(500),IOBND(500)
1639.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
1640.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1641.      2      IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVRQ,ITFRQ,ITSINV,IFFEZ,
1642.      3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1643.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1644.      DP = X(IOBJ) + ZTOLZE
1645.      IVAL = IDINT(DP)
1646.      IF (DP .LT. 0.) IVAL = IVAL - 1

```



```

1647.      IF (MSTAT .EQ. QN) .OR. (IVAL .LE. INCVAL) GO TO 2000
1648.      C
1649.      C          COMPUTE INTEGER AND FRACTIONAL PARTS OF EACH BASIC VAR.
1650.      C
1651.      DO 100 I=1,NROW
1652.      IPART(I) = IDINT(X(I) + ZTOLZE)
1653.      NTEMP(1) = IPART(I)
1654.      100 DFPAKT(I) = X(I) - FLOAT(NTEMP(1))
1655.      C
1656.      C          CHECK FOR ALL-INTEGER SOLUTION
1657.      C
1658.      DO 200 I=1,NROW
1659.      IF (JH(I) .LE. NROW) GO TO 200
1660.      IF (DFPAKT(I) .GE. ZTOLZE) RETURN
1661.      200 CONTINUE
1662.      C
1663.      C          SOLUTION ALL-INTEGER: INSTALL AS NEW INCUMBENT
1664.      C
1665.      IFEAS = 1
1666.      INCVAL = IVAL
1667.      DO 1000 J=1,NCOL
1668.      IF (KINBAS(J)) 600,700,1000
1669.      600 INCUMB(J) = IDINT(XUB(J))
1670.      GO TO 1000
1671.      700 INCUMB(J) = IDINT(XLB(J))
1672.      1000 CONTINUE
1673.      C
1674.      DO 1100 I=1,NROW
1675.      ICOL = JH(I)
1676.      INCUMB(ICOL) = IPART(I)
1677.      1100 CONTINUE
1678.      C
1679.      C          CURRENT PROBLEM NO LONGER OF INTEREST
1680.      C
1681.      2000 MSTAT = QI
1682.      RETURN
1683.      END
1684.      SUBROUTINE PENLTS
1685.      C
1686.      C          COMPUTES TOMLIN'S IMPROVED UP- AND DOWN-PENALTIES AND THE
1687.      C          GCMORY PENALTY FOR EACH NONINTEGER BASIC VARIABLE. ALSO CHECKS
1688.      C          FOR FORCED BRANCHES ON BOTH BASIC AND NONBASIC VARIABLES. IN
1689.      C          THE ABSENCE OF FORCED BRANCHES ON BASIC VARIABLES, THE
1690.      C          BRANCHING VARIABLE IS CHOSEN TO BE THE ONE WITH LARGEST
1691.      C          ASSOCIATED UP- OR DOWN-PENALTY. THE FORWARD BRANCH IS MADE IN
1692.      C          THE DIRECTION OPPOSITE TO THE MAXIMUM PENALTY. THE NODE
1693.      C          CORRESPONDING TO THE BRANCH IN THE SAME DIRECTION AS THE
1694.      C          MAXIMUM PENALTY IS ADDED TO THE LIST, TO BE EXAMINED LATER.
1695.      C          THE BRANCHING PROCESS ITSELF IS CARRIED OUT IN SUBROUTINE
1696.      C          BRANCH, WHICH IS CALLED FROM SUBROUTINE PENLTS.
1697.      C
1698.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1699.      1      INTEGER*4 (I-N,Q)
1700.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1701.      INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
1702.      DOUBLE PRECISION E(1000)
1703.      REAL A(500)
1704.      REAL PU(60),PD(60),PG(60)
1705.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1706.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1707.      2      QE,QF,QL,QO,QR,QM,QG,NTMAX

```

AD-A033 948

STANFORD UNIV CALIF DEPT OF OPERATIONS RESEARCH  
COMPUTER PROGRAMS FOR DECOMPOSITION IN INTEGER PROGRAMMING.(U)  
SEP 76 G A KOCHMAN

F/G 9/2

N00014-76-C-0418

UNCLASSIFIED

TR-71

NL

2 OF 2

AD  
A033948



END

DATE  
FILMED

2-77

```

1708.      COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1709.      1      NPIVOT,IPTYPE,RCGST,IFEAS,IPART(122),INCUMB(122),
1710.      2      IVBND(500),IVID(500),IOBND(500)
1711.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
1712.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1713.      2      IOBJ,IRCWP,IVIN,IVOJT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
1714.      3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1715.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1716.      DO 10 I=1,NROW
1717.      IF (DFPART(I) .LT. ZTOLZE) GO TO 5
1718.      PU(I) = 1.E6
1719.      PD(I) = 1.E6
1720.      PG(I) = 1.E6
1721.      GC TO 10
1722.      5 PU(I) = 0.
1723.      PD(I) = 0.
1724.      PG(I) = 0.
1725.      10 CCNTINUE
1726.      C
1727.      DO 1000 J=1,NCOL
1728.      IF (KINBAS(J) .GT. 0) GO TO 1000
1729.      IF ((XUB(J) - XLB(J)) .LE. ZTOLZE) GO TO 1000
1730.      CALL UNPACK(J)
1731.      CALL FTRAN(J)
1732.      IF (KINBAS(J) .EQ. 0) GO TO 30
1733.      DO 20 I=1,NROW
1734.      20 Y(I) = -Y(I)
1735.      C
1736.      C      CHECK FOR FORCED BRANCH ON XJ
1737.      30 IF (J .LE. NROW) GC TO 50
1738.      DP = X(IOBJ) - Y(IOBJ) + ZTOLZE
1739.      IVAL = IDINT(DP)
1740.      IF (DP .LT. 0.) IVAL = IVAL - 1
1741.      IF (IVAL .GT. INCVAL) GO TO 50
1742.      IDIR = 2*KINBAS(J) + 1
1743.      IF (ICIR .EQ. -1) REVBND = SNGL(XUB(J))
1744.      IF (ICIR .EQ. 1) REVBND = SNGL(XLB(J))
1745.      ICOL = J
1746.      CALL BRANCH
1747.      GO TO 1000
1748.      C
1749.      50 DO 500 I=1,NROW
1750.      IF (JH(I) .LE. NROW) GO TO 500
1751.      IF (DFPART(I) .LT. ZTOLZE) GO TO 500
1752.      C
1753.      C      COMPUTE UP PENALTY FOR X(I), XJ
1754.      100 IF (Y(I) .GT. -ZTOLPV) GO TO 200
1755.      DE = Y(IOBJ)*(DFPART(I) - 1.)/Y(I)
1756.      IF (DE .LT. Y(IOBJ)) DE = Y(IOBJ)
1757.      IF (DE .LT. PU(I)) PU(I) = DE
1758.      GO TO 300
1759.      C
1760.      C      COMPUTE DOWN PENALTY FOR X(I), XJ
1761.      200 IF (Y(I) .LT. ZTOLPV) GO TO 300
1762.      DE = Y(IOBJ)*DFPART(I)/Y(I)
1763.      IF (DE .LT. Y(IOBJ)) DE = Y(IOBJ)
1764.      IF (DE .LT. PD(I)) PD(I) = DE
1765.      C
1766.      C      COMPUTE GOMORY PENALTY FOR X(I), XJ
1767.      300 DP = CABS(Y(I))
1768.      NTEMP(1) = IDINT(DP)

```



```

1769.      DP = CP - FLOAT(NTEMP(1))
1770.      IF ((CP .LE. ZTOLZE) .OR. (DP .GE. 1.-ZTOLZE)) GO TO 500
1771.      IF (Y(I) .LT. 0.) DP = 1. - DP
1772.      IF (DP .GT. DFPART(I)) GO TO 310
1773.      DE = Y(I0BJ)*DFPART(I)/DP
1774.      GO TO 320
1775.      31C DE = Y(I0BJ)*(1. - DFPART(I))/(1. - DP)
1776.      32C IF (DE .LT. PG(I)) PG(I) = DE
1777.      C
1778.      5CC CONTINUE
1779.      10CC CONTINUE
1780.      C
1781.      C      COMPUTE LARGEST GOMORY PENALTY AND TEST FOR FATHOMING
1782.      PEN = 0.
1783.      DO 20CC I=1,NROW
1784.      IF (JH(I) .LE. NROW) GO TO 2000
1785.      IF (PG(I) .GT. PEN) PEN = PG(I)
1786.      20CC CONTINUE
1787.      DP = X(I0BJ) - PEN + ZTOLZE
1788.      IVAL = IDINT(DP)
1789.      IF (DP .LT. 0.) IVAL = IVAL - 1
1790.      IF (IVAL .GT. INCVAL) GO TO 3000
1791.      IDIR = 0
1792.      RETURN
1793.      C
1794.      C      PROBLEM NOT FATHOMED: CHECK FOR FORCED BRANCHES ON X(I)
1795.      30CC NTEMP(2) = 0
1796.      DO 39CC I=1,NROW
1797.      IF (JH(I) .LE. NROW) GO TO 3900
1798.      IF (PU(I) .GT. PD(I)) GO TO 3600
1799.      DP = X(I0BJ) - PD(I) + ZTOLZE
1800.      NTEMP(1) = IDINT(DP)
1801.      IF (DP .LT. 0.) NTEMP(1) = NTEMP(1) - 1
1802.      IF (NTEMP(1) .GT. INCVAL) GO TO 3900
1803.      C      FORCED BRANCH UP ON X(I)
1804.      IVAL = NTEMP(1)
1805.      IDIR = -1
1806.      NTEMP(1) = IPART(I) + 1
1807.      GO TO 3700
1808.      C
1809.      36CC DP = X(I0BJ) - PU(I) + ZTOLZE
1810.      NTEMP(1) = IDINT(DP)
1811.      IF (DP .LT. 0.) NTEMP(1) = NTEMP(1) - 1
1812.      IF (NTEMP(1) .GT. INCVAL) GO TO 3900
1813.      C      FORCED BRANCH DOWN ON X(I)
1814.      IVAL = NTEMP(1)
1815.      IDIR = 1
1816.      NTEMP(1) = IPART(I)
1817.      37CC IROWP = I
1818.      ICOL = JH(IROWP)
1819.      REVBNC = FLOAT(NTEMP(1))
1820.      NTEMP(2) = 1
1821.      CALL BRANCH
1822.      39CC CONTINUE
1823.      IF (NTEMP(2) .GT. C) GO TO 5000
1824.      C
1825.      C      NO FORCED BRANCHES: CHOOSE BRANCHING VAR. AND DIRECTION
1826.      PEN = 0.
1827.      IROWP = 0
1828.      C      DETERMINE BASIC VAR. X(IROWP) WITH MAX. UP- OR DOWN-PENALTY
1829.      DO 49CC I=1,NROW

```

```

1830.      IF (JF(I) .LE. NROW) GO TO 4900
1831.      IF (PU(I) .GT. PD(I)) GO TO 4600
1832.      IF (PC(I) .LE. PEN) GO TO 4900
1833.      PEN = PD(I)
1834.      IROWP = I
1835.      IDIR = -1
1836.      NTEMP(1) = IPART(I) + 1
1837.      REVBNC = FLOAT(NTEMP(1))
1838.      GO TO 4900
1839. 4600 IF (PU(I) .LE. PEN) GO TO 4900
1840.      PEN = PU(I)
1841.      IROWP = I
1842.      IDIR = 1
1843.      NTEMP(1) = IPART(I)
1844.      REVBNC = FLOAT(NTEMP(1))
1845. 4900 CCNTINUE
1846.      IF (IROWP .GT. 0) GO TO 4950
1847. C      EACH UP- AND DOWN-PENALTY .EQ. 0. (DUAL-DEGENERACY). CHOOSE
1848. C      ANY NONINTEGER BASIC VAR. AS BRANCHING VAR.
1849.      DO 4910 IROWP=1,NRGW
1850.      IF (JF(IROWP) .LE. NROW) GO TO 4910
1851.      IF (DFPART(IROWP) .GE. ZTOLZE) GO TO 4920
1852. 4910 CONTINUE
1853. 4920 PEN = PU(IROWP)
1854.      IDIR = 1
1855.      NTEMP(1) = IPART(IROWP)
1856.      REVBNC = FLOAT(NTEMP(1))
1857. 4950 ICOL = JH(IROWP)
1858.      DP = X(IOBJ) - PEN + ZTOLZE
1859.      NTEMP(1) = IDINT(DP)
1860.      IF (DP .LT. 0.) NTEMP(1) = NTEMP(1) - 1
1861.      IF (IVAL .GT. NTEMP(1)) IVAL = NTEMP(1)
1862. C      BRANCH ON CHOSEN VAR.
1863.      CALL BRANCH
1864. 5000 IF (ICIR .EQ. -1) IPTYPE = 0
1865.      IF (ICIR .EQ. 1) IPTYPE = -1
1866.      RETURN
1867.      END
1868.      SUBROUTINE BRANCH
1869. C
1870. C      BRANCHES ON VARIABLE X(ICOL) AS DETERMINED IN SUBROUTINE PENLTS
1871. C
1872.      IMPLICIT REAL*4 (A,C,E-H,G,P,R-W,Z), REAL*8 (B,D,X,Y),
1873. 1      INTEGER*4 (I-N,Q)
1874.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1875.      INTEGER*2 IPART,INCUMB,IVBND,IVID,IOBND
1876.      DOUBLE PRECISION E(1000)
1877.      REAL A(500)
1878.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSK,QRO,QMA,QBA,QFI,
1879. 1      QEO,CBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1880. 2      QE,QH,QL,QO,QR,QM,QG,NTMAX
1881.      COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1882. 1      NPivot,IPTYPE,RCOST,IFEAS,IPART(122),INCUMB(122),
1883. 2      IVBND(500),IVID(500),IOBND(500)
1884.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
1885. 1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1886. 2      IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFE2,
1887. 3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1888. 4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1889. C
1890. C      ICOL INDEXES BRANCHING VARIABLE CHOSEN

```



```

1891. C IDIR INDICATES BRANCHING DIRECTION CHOSEN
1892. C
1893. C ADD OPPOSITE DIRECTION TO LIST
1894. LISTL = LISTL + 1
1895. IF (ICIR .EQ. -1) IVBND(LISTL) = IDINT(XLB(ICOL) + ZTOLZE)
1896. IF (ICIR .EQ. 1) IVBND(LISTL) = IDINT(XUB(ICOL) + ZTOLZE)
1897. IVID(LISTL) = IDIR*ICOL
1898. IOBND(LISTL) = IVAL
1899. C
1900. C REVISE BOUNDS ON BRANCHING VARIABLE FOR FORWARD DIRECTION
1901. IF (ICIR .EQ. -1) XLB(ICOL) = DBLE(REVBND)
1902. IF (ICIR .EQ. 1) XUB(ICOL) = DBLE(REVBND)
1903. RETURN
1904. END
1905. SUBROUTINE BKTRAK
1906. C
1907. C BACKTRACKS TO SELECT A PROMISING (UNFATHOMED) NODE FROM THE
1908. C LIST OF STORED NODES. LAST-IN-FIRST-OUT (LIFO) SELECTION RULE
1909. C IS EMPLOYED.
1910. C
1911. IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1912. 1 INTEGER*4 (I-N,Q)
1913. INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1914. INTEGER*2 IPART,INCUMB,IVBND,IVID,IOBND
1915. DOUBLE PRECISION E(1000)
1916. REAL A(500)
1917. COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1918. 1 QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1919. 2 QE,QH,QL,QO,QR,QM,QG,NTMAX
1920. COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1921. 1 NPIVCT,IPTYPE,RCOST,IFEAS,IPART(122),INCUMB(122),
1922. 2 IVBND(500),IVID(500),IOBND(500)
1923. COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
1924. 1 DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1925. 2 IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
1926. 3 JCPLP,NROW,ACOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1927. 4 IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1928. NTEMP(3) = 0
1929. C IF LIST IS EMPTY, RETURN (COMPUTATIONS COMPLETED)
1930. 50 IF (LISTL .EQ. 0) RETURN
1931. IF (IOBND(LISTL) .LE. INCVAL) GO TO 2000
1932. C
1933. C GET NEXT NODE FROM LIST
1934. ICOL = IVID(LISTL)
1935. IF (ICOL .LT. 0) GO TO 100
1936. C
1937. NTEMP(1) = IDINT(XLB(ICOL) + ZTOLZE)
1938. NTEMP(2) = IVBND(LISTL)
1939. XLB(ICOL) = XUB(ICOL) + 1.
1940. XUB(ICOL) = FLOAT(NTEMP(2))
1941. IF (KINBAS(ICOL) .GT. 0) GO TO 1000
1942. KINBAS(ICOL) = 0
1943. NTEMP(3) = 1
1944. GO TO 1000
1945. C
1946. 100 ICOL = -ICOL
1947. NTEMP(1) = IDINT(XUB(ICOL) + ZTOLZE)
1948. NTEMP(2) = IVBND(LISTL)
1949. XUB(ICOL) = XLB(ICOL) - 1.
1950. XLB(ICOL) = FLOAT(NTEMP(2))
1951. IF (KINBAS(ICOL) .GT. 0) GO TO 1000

```



```

1952.      KINBAS(ICOL) = -1
1953.      NTEMP(3) = 1
1954.      C
1955.      1000 IVID(LISTL) = -IVID(LISTL)
1956.      IVBND(LISTL) = NTEMP(1)
1957.      IOBND(LISTL) = -10000
1958.      C
1959.      C      UPDATE X
1960.      IF (NTEMP(3) .EQ. C) RETURN
1961.      CALL SHIFTR(1,3)
1962.      DO 9000 J=1,NCOL
1963.      IF (KINBAS(J)) 8600,8700,9000
1964.      8600 DE = XUB(J)
1965.      GO TO 8750
1966.      8700 DE = XLB(J)
1967.      8750 LL = LA(J)
1968.      KK = LA(J+1) - 1
1969.      DO 8800 I=LL,KK
1970.      IR = IA(I)
1971.      8800 Y(IR) = Y(IR) - A(I)*DE
1972.      9000 CONTINUE
1973.      CALL FTRAN(1)
1974.      CALL SHIFTR(3,2)
1975.      RETURN
1976.      C
1977.      C      NODE FATHOMED:  UPDATE VAR.  BOUNDS AND BACKTRACK AGAIN
1978.      2000 ICOL = IVID(LISTL)
1979.      IF (ICOL .LT. 0) GO TO 2100
1980.      C
1981.      NTEMP(1) = IVBND(LISTL)
1982.      IF (KINBAS(ICOL)) 2010,2050,2050
1983.      2010 NTEMP(3) = 1
1984.      DP = XUB(ICOL) - XLB(ICOL)
1985.      DY = FLOAT(NTEMP(1)) - XUB(ICOL)
1986.      IF (DP .LT. DY) KINBAS(ICOL) = 0
1987.      2050 XUB(ICOL) = FLOAT(NTEMP(1))
1988.      GO TO 3000
1989.      C
1990.      2100 ICOL = -ICOL
1991.      NTEMP(1) = IVBND(LISTL)
1992.      IF (KINBAS(ICOL)) 2150,2110,2150
1993.      2110 NTEMP(3) = 1
1994.      DY = XLB(ICOL) - FLOAT(NTEMP(1))
1995.      DP = XUB(ICOL) - XLB(ICOL)
1996.      IF (DP .LT. DY) KINBAS(ICOL) = -1
1997.      2150 XLB(ICOL) = FLOAT(NTEMP(1))
1998.      C
1999.      3000 LISTL = LISTL - 1
2000.      GO TO 50
2001.      C
2002.      END
2003.      SUBROUTINE ALOCTE(IPAR,IRHS)
2004.      C
2005.      C      SOLVES MASTER PROBLEM (CONCAVE-SEPARABLE INTEGER PROGRAM);
2006.      C      REALLOCATES LINKING RESOURCE AMONG UNFIXED SUBPROBLEMS
2007.      C*****DESCRIPTION OF PARAMETERS*****
2008.      C      IPAR = PARAMETER INDICATING WHETHER LAST BRANCH WAS IN FORWARD
2009.      C      (IPAR = 1) OR BACKTRACKING DIRECTION (IPAR = 0).
2010.      C      IRHS = AMOUNT OF LINKING RESOURCE LEFT (USED WHEN IPAR = 1)
2011.      C*****
2012.      C

```

```

2013. C      WHEN IPAR = 1, MASTER PROBLEM REOPTIMIZED BY REALLOCATING IRHS,
2014. C      STARTING FROM PREVIOUSLY OPTIMAL SOLUTION TO MASTER PROBLEM.
2015. C      WHEN IPAR = 0, MASTER PROBLEM RESOLVED FROM SCRATCH (I.E., ALL
2016. C      B(J) SET TO LOWER BOUNDS IBLB(J) INITIALLY.)
2017. C
2018.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
2019.      1      INTEGER*4 (I-N,Q)
2020.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
2021.      INTEGER*2 IPART,INCLMB,IVBND,IVID,IQBND
2022.      INTEGER*2 IZSTAR,IBID,IBBND,IZBND
2023.      DOUBLE PRECISION E(1000)
2024.      REAL A(500)
2025.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
2026.      1      QEQ,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
2027.      2      QE,QH,QL,QU,QR,QM,QG,NTMAX
2028.      COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
2029.      1      NPIVOT,IPTYPE,RCOST,IFEAS,IPART(122),INCUMB(122),
2030.      2      IVBND(500),IVID(500),IOBND(500)
2031.      COMMON/BLOCK3/ RLAMDA(50,10),ZNAUT(10),ZJBAR(10),NSUBS,IZINC,
2032.      1      IZBAR,IDOM,IBLL,NJFIX,JFIXSM,IB,JFIX(10),NSOL(10),NLAMDA(10),
2033.      2      KSEG(10),IBSTAR(200,10),IBUB(10),IBLB(10),IBP(51,10),
2034.      3      IBUSEC(200,10),IZSTAR(200,10),IBID(200),IBBND(200),IZBND(200)
2035.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
2036.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
2037.      2      IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
2038.      3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
2039.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
2040. C      TEST WHETHER LAST BRANCH WAS IN FORWARD OR BACKTRACKING DIREC-
2041. C      TICN
2042.      IF (IPAR .EQ. 0) GO TO 50
2043. C      LAST BRANCH WAS IN FORWARD DIRECTION:  INITIALIZE B(J) TO
2044. C      PREVIOUSLY OPTIMAL SOLUTION
2045.      DO 10 J=1,NSUBS
2046.      LAST = NSOL(J) + 1
2047.      IF (JFIX(J) .GT. 0) LAST = JFIX(J)
2048.      10 NTEMP(J) = IBSTAR(LAST,J)
2049.      GO TO 200
2050. C
2051. C      LAST BRANCH WAS IN BACKTRACKING DIRECTION
2052. C      SET ALL B(J) = IBLB(J) AND IRHS = AMT OF RESOURCE LEFT
2053.      50 INCVAL = -10000
2054.      IRHS = IB
2055.      DO 90 J=1,NSUBS
2056.      50 IRHS = IRHS - IBLB(J)
2057.      IF (IRHS .LT. 0) RETURN
2058. C
2059.      DO 130 J=1,NSUBS
2060.      IF (JFIX(J) .GT. 0) GO TO 130
2061.      ZJBAR(J) = ZNAUT(J)
2062.      LAST = NLAMDA(J) + 1
2063.      DO 110 K=2,LAST
2064.      KM1 = K - 1
2065.      IF (IBP(K,J) .GT. IBLB(J)) GO TO 120
2066.      110 ZJBAR(J) = ZJBAR(J) + RLAMDA(KM1,J)*(IBP(K,J) - IBLB(J))
2067.      KSEG(J) = LAST
2068.      GO TO 130
2069.      120 KSEG(J) = KM1
2070.      ZJBAR(J) = ZJBAR(J) + RLAMDA(KM1,J)*(IBLB(J) - IBP(KM1,J))
2071.      130 NTEMP(J) = IBLB(J)
2072. C
2073. C      ALLOCATE:  GET LARGEST UP-SLOPE

```



```

2074.      200 RLMAX = -1.
2075.      DO 220 J=1,NSUBS
2076.      IF (JFIX(J) .GT. 0) GO TO 220
2077.      IF (NTEMP(J) .EQ. IBUB(J)) GO TO 220
2078.      K = KSEG(J)
2079.      IF (RLMAX .GE. RLAMDA(K,J)) GO TO 220
2080.      RLMAX = RLAMDA(K,J)
2081.      JSTORE = J
2082.      220 CONTINUE
2083.      IF (RLMAX .LE. -0.5) GO TO 1100
2084.      C          INCREMENT CORRESPONDING B(J)
2085.      K = KSEG(JSTORE)
2086.      KPI = K + 1
2087.      IDelta = IBP(KPI,JSTORE) - NTEMP(JSTORE)
2088.      IF (IBP(KPI,JSTORE) .GT. IBUB(JSTORE))
2089.      1          IDelta = IBUB(JSTORE) - NTEMP(JSTORE)
2090.      IRHS = IRHS - IDelta
2091.      NTEMP(JSTORE) = NTEMP(JSTORE) + IDelta
2092.      ZJBAR(JSTORE) = ZJBAR(JSTORE) + RLAMDA(K,JSTORE)*IDelta
2093.      IF (NTEMP(JSTORE) .EQ. IBP(KPI,JSTORE)) KSEG(JSTORE)=KSEG(JSTORE)+1
2094.      IF (IRHS .LE. 0) GO TO 1000
2095.      GO TO 200
2096.      C
2097.      C          END OF ALLOCATION
2098.      1000 NTEMP(JSTORE) = NTEMP(JSTORE) + IRHS
2099.      ZJBAR(JSTORE) = ZJBAR(JSTORE) + RLAMDA(K,JSTORE)*IRHS
2100.      K = KSEG(JSTORE)
2101.      IF (NTEMP(JSTORE) .LT. IBP(K,JSTORE)) KSEG(JSTORE)=K-1
2102.      C          INSTALL OPTIMAL ALLOCATION
2103.      1100 ZBAR = ZTOLZE
2104.      DO 1110 J=1,NSUBS
2105.      IF (JFIX(J) .GT. 0) GO TO 1105
2106.      LAST = NSOL(J) + 1
2107.      IBSTAR(LAST,J) = NTEMP(J)
2108.      ZBAR = ZBAR + ZJBAR(J)
2109.      GO TO 1110
2110.      1105 K = JFIX(J)
2111.      IDelta = IZSTAR(K,J)
2112.      ZBAR = ZBAR + IDelta
2113.      1110 CONTINUE
2114.      INCVAL = INT(ZBAR)
2115.      IF (ZBAR .LT. 0.) INCVAL = INCVAL - 1
2116.      RETURN
2117.      END
2118.      SUBROUTINE DOMCHK(J,K)
2119.      C
2120.      C          COMPARES CURRENT ALLOCATION (FROM SUBROUTINE ALOCTE) WITH
2121.      C          PREVIOUSLY EXAMINED ALLOCATIONS IN EACH UNFIXED SUBPROBLEM
2122.      C          TO DETERMINE IF OPT. SOL. CORRESPONDING TO CURRENT ALLOCATION
2123.      C          IS ALREADY KNOWN FOR ANY SUBPROBLEM
2124.      C*****DESCRIPTION OF PARAMETERS*****
2125.      C          J = PARAMETER INDEXING FIRST SUBPROBLEM FOUND FOR WHICH OPTIMAL
2126.      C          SOLUTION CORRESPONDING TO CURRENT ALLOCATION IS ALREADY
2127.      C          KNOWN (OUTPUT)
2128.      C          K = PARAMETER INDEXING THE PREVIOUSLY EXAMINED ALLOCATION IN
2129.      C          SUBPROBLEM J WHICH LEADS TO SAME OPT. SOL. AS WOULD
2130.      C          CURRENT ALLOCATION (OUTPUT)
2131.      C*****
2132.      C
2133.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
2134.      1      INTEGER*4 (I-N,Q)

```



```

2135.      INTEGER*2 IZSTAR,IBID,IBBND,IZBND
2136.      COMMON/BLOCK3/ RLAMDA(50,10),ZNAUT(10),ZJBAR(10),NSUBS,IZINC,
2137.      IIZBAR,IDOM,IBLL,NJFIX,JFIXSM,IB,JFIX(10),NSOL(10),NLAMDA(10),
2138.      2KSEG(10),IBSTAR(200,10),IBUB(10),IBLB(10),IBP(51,10),
2139.      3IBUSED(200,10),IZSTAR(200,10),IBID(200),IBBND(200),IZBND(200)
2140.      DO 1000 J=1,NSUBS
2141.      IF (JFIX(J) .GT. 0) GO TO 1000
2142.      IF (NSOL(J) .EQ. 0) GO TO 1000
2143.      L = NSOL(J)
2144.      LPI = L + 1
2145.      DO 500 K=1,L
2146.      C          TEST ALLOCATION AGAINST UPPER BOUNDS OF
2147.      C          BIJ IN KTH SOL. OF SUB J
2148.      IF (IBSTAR(LPI,J) .GT. IBSTAR(K,J)) GO TO 500
2149.      C          TEST AGAINST LOWER BOUNDS
2150.      IF (IBSTAR(LPI,J) .LT. IBUSED(K,J)) GO TO 500
2151.      C
2152.      C          ALLOCATION DOMINATED
2153.      IDOM = 1
2154.      RETURN
2155.      C
2156.      500 CONTINUE
2157.      1000 CONTINUE
2158.      C
2159.      C          ALLOCATION NOT DOMINATED
2160.      IDOM = 0
2161.      RETURN
2162.      END
2163.      SUBROUTINE LDDATA(J)
2164.      C
2165.      C          LOADS DATA FOR SUBPROBLEM J INTO APPROPRIATE STORAGE LOCATIONS
2166.      C          FOR SOLUTION BY SUBROUTINES NORMAL (LP) AND BANDB (ILP)
2167.      C
2168.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
2169.      1      INTEGER*4 (I-N,Q)
2170.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
2171.      INTEGER*2 IZSTAR,IBID,IBBND,IZBND
2172.      INTEGER*2 IAS,LAS
2173.      DOUBLE PRECISION E(1000)
2174.      REAL A(500)
2175.      COMMON/BLOCK3/ RLAMDA(50,10),ZNAUT(10),ZJBAR(10),NSUBS,IZINC,
2176.      IIZBAR,IDOM,IBLL,NJFIX,JFIXSM,IB,JFIX(10),NSOL(10),NLAMDA(10),
2177.      2KSEG(10),IBSTAR(200,10),IBUB(10),IBLB(10),IBP(51,10),
2178.      3IBUSED(200,10),IZSTAR(200,10),IBID(200),IBBND(200),IZBND(200)
2179.      COMMON/BLOCK4/ XLBS(50,10),XUBS(50,10),AMATRS(1000),RHS(20,10),
2180.      1NROWS(10),NCOLS(10),JFIRST(11),IAS(1000),LAS(50,10)
2181.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
2182.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
2183.      2      IOBJ,IRCWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
2184.      3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
2185.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
2186.      NCOL = NCOLS(J)
2187.      NROW = NROWS(J)
2188.      NELEM = JFIRST(J+1) - JFIRST(J)
2189.      LL = JFIRST(J) - 1
2190.      C          LOAD A,IA,LA,XLB,XUB,B,JH,KINBAS
2191.      DO 10 I=1,NELEM
2192.      A(I) = AMATRS(LL+I)
2193.      10 A(I) = IAS(LL+I)
2194.      DO 20 K=1,NCOL
2195.      LA(K) = LAS(K,J)

```

```

2196.      XLB(K) = XLBS(K,J)
2197. 20 XUB(K) = XUBS(K,J)
2198.      LA(NCCL+1) = NELEM + 1
2199.      B(1) = 0.
2200.      LL = NSOL(J) + 1
2201.      B(2) = FLOAT(IBSTAR(LL,J))
2202.      LL = NROW - 2
2203.      DO 40 I=1,LL
2204. 40 B(I+2) = DBLE(RHS(I,J))
2205.      DO 50 I=1,NROW
2206.      JH(I) = I
2207. 50 KINBAS(I) = 1
2208.      LL = NROW + 1
2209.      DO 60 I=LL,NCOL
2210. 60 KINBAS(I) = 0
2211.      ITCNT = 0
2212.      INVFRQ = NROW
2213.      RETURN
2214.      END
2215.      SUBROUTINE GETLBD(J)
2216. C
2217. C      COMPARES CURRENT ALLOCATION IN SUBPROBLEM J WITH PREVIOUSLY
2218. C      EXAMINED ALLOCATIONS IN ORDER TO DETERMINE AN INITIAL LOWER
2219. C      BOUND ON THE MAXIMAL OBJECTIVE VALUE IN SUBPROBLEM J,
2220. C      CORRESPONDING TO CURRENT ALLOCATION
2221. C
2222.      IMPLICIT REAL*4 (A,C,E-H,G,P,R-W,Z), REAL*8 (B,D,X,Y),
2223. 1      INTEGER*4 (I-N,Q)
2224.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
2225.      INTEGER*2 IPART,INCUMB,IVBND,IVID,I0BND
2226.      INTEGER*2 IZSTAR,IBID,IBBND,IZBND
2227.      DOUBLE PRECISION E(1000)
2228.      REAL A(500)
2229.      COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
2230. 1      NPVGT,IPTYPE,RCOST,IFEAS,IPART(122),INCUMB(122),
2231. 2      IVBND(500),IVID(500),I0BND(500)
2232.      COMMON/BLOCK3/ RLAMDA(50,10),ZNAUT(10),ZJBAR(10),NSUBS,IZINC,
2233. 1      IZBAR,IDUM,IBLL,NJFIX,JFIXSM,IB,JFIX(10),NSOL(10),NLAMDA(10),
2234. 2      KSEG(10),IBSTAR(200,10),IBUB(10),IBLB(10),IBP(51,10),
2235. 3      IBUSEC(200,10),IZSTAR(200,10),IBID(200),IBBND(200),IZBND(200)
2236.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
2237. 1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
2238. 2      IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
2239. 3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
2240. 4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
2241.      INCUMB(1) = -10000
2242.      KSTORE = 0
2243.      LL = NSOL(J)
2244.      IF (LL .EQ. 0) RETURN
2245.      NEW = LL + 1
2246. C      SEARCH PREVIOUS ALLOCATIONS FOR BEST LOWER BOUND ON ZJ
2247. C      DO 100 K=1,LL
2248. C      IF (INCUMB(1) .GE. IZSTAR(K,J)) GO TO 100
2249. C      IF (IBUSED(K,J) .GT. IBSTAR(NEW,J)) GO TO 100
2250. C      BIJ* >= KTH BIJ: UPDATE INCUMB(1)
2251. C      INCUMB(1) = IZSTAR(K,J)
2252. C      KSTORE = K
2253. 100 CONTINUE
2254. C
2255. C      KSTORE INDEXES THE PREVIOUSLY EXAMINED ALLOCATION GIVING THE
2256. C      GREATEST LOWER BOUND ON ZJ

```



```

2257.      IF (KSTORE .EQ. 0) RETURN
2258.      INCUMB(2) = IBSTAR(NEW,J) - IBUSED(KSTORE,J)
2259.      DO 210 K=3,NCOL
2260. 210 INCUMB(K) = 0
2261.      IFEAS = 1
2262.      RETURN
2263.      END
2264.      SUBROUTINE SAVER(J)
2265. C
2266. C      STORES ALLOCATION AND SOLUTION INFORMATION FOR SUBPROBLEM J
2267. C
2268.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-h,Z), REAL*8 (B,D,X,Y),
2269. 1      INTEGER*4 (I-N,Q)
2270.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
2271.      INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
2272.      INTEGER*2 IZSTAR,IBID,IBBND,IZBND
2273.      DOUBLE PRECISION E(1000)
2274.      REAL A(500)
2275.      COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
2276. 1      NPVCT,IPTYPE,RCOST,IFEAS,IPART(122),INCUMB(122),
2277. 2      IVBND(500),IVID(500),IOBND(500)
2278.      COMMON/BLOCK3/ RLAMDA(50,10),ZNAUT(10),ZJBAR(10),NSUBS,IZINC,
2279. 1      IZBAR,IDOM,IBLL,NJFIX,JFIXSM,IB,JFIX(10),NSOL(10),NLAMDA(10),
2280. 2      KSEG(10),IBSTAR(200,10),IBUB(10),IBLB(10),IBP(51,10),
2281. 3      IBUSEC(200,10),IZSTAR(200,10),IBIC(200),IBBND(200),IZBND(200)
2282.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
2283. 1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
2284. 2      IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
2285. 3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
2286. 4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
2287. C      TEST WHETHER SUBPROBLEM J WAS FEASIBLE (IFEAS = 1)
2288.      IF (IFEAS .EQ. 1) GO TO 90
2289.      INCVAL = -10000
2290.      INCUMB(1) = -10000
2291.      INCUMB(2) = 10000
2292. C
2293.      SO LL = NSOL(J)
2294.      NEW = LL + 1
2295.      IBUSED(NEW,J) = IBSTAR(NEW,J) - INCUMB(2)
2296.      IF (LL .EQ. 0) GO TO 1000
2297. C      COMPARE NEW SOLUTION IN SUB J WITH PREVIOUS SOLUTIONS
2298.      DO 500 K=1,LL
2299.      IF (IBUSED(NEW,J) .NE. IBUSED(K,J)) GO TO 500
2300. C      NEW SOLUTION SAME AS KTH: REWRITE OVER KTH REGION
2301.      IBSTAR(K,J) = IBSTAR(NEW,J)
2302.      NTEMP(2) = K
2303.      RETURN
2304.      SOO CCNTINUE
2305. C
2306. C      NEW SOLUTION DIFFERS FROM ALL PREVIOUS SOLUTIONS
2307.      1000 NSOL(J) = NEW
2308.      IZSTAR(NEW,J) = INCUMB(1)
2309.      NTEMP(2) = NEW
2310.      RETURN
2311.      END
2312.      SUBROUTINE DMBRAN(J,K)
2313. C
2314. C      BRANCHES ON ALLOCATION VARIABLES BIJ IN SUBPROBLEM J
2315. C*****DESCRIPTION OF PARAMETERS*****
2316. C      J = PARAMETER INDEXING SUBPROBLEM IN WHICH BRANCHING IS DONE
2317. C      K = PARAMETER INDEXING A PREVIOUSLY EXAMINED ALLOCATION IN

```



```

2318. C SUBPROBLEM J. THE CURRENT ALLOCATION IN SUBPROBLEM J IS TO
2319. C BE FIXED AT THE KTH LEVEL FOLLOWING THE BRANCHING PROCEDURE
2320. C*****
2321. C
2322. C IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
2323. C 1 INTEGER*4 (I-N,Q)
2324. C INTEGER*2 JH,KINBAS,LA,LE,IA,IE
2325. C INTEGER*2 IZSTAR,IBID,IBBND,IZBND
2326. C DCUBLE PRECISION E(1000)
2327. C REAL A(500)
2328. C COMMON/BLOCK3/ RLAMDA(50,10),ZNAUT(10),ZJBAR(10),NSUBS,IZINC,
2329. C IZBAR,IDUM,IBLL,NJFIX,JFIXSM,IB,JFIX(10),NSOL(10),NLAMDA(10),
2330. C 2KSEG(10),IBSTAR(200,10),IBUB(10),IBLB(10),IBP(51,10),
2331. C 3IBUSED(200,10),IZSTAR(200,10),IBID(200),IBBND(200),IZBND(200)
2332. C COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
2333. C 1 DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
2334. C 2 IUBJ,IRGWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
2335. C 3 JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
2336. C 4 IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
2337. C NTEMP(1) = IBUSED(K,J)
2338. C IF (IBSTAR(K,J) .GE. IBUB(J)) GO TO 200
2339. C ADD UP BRANCH ON BIJ TO LIST
2340. C IBLL = IBLL + 1
2341. C IBID(IBLL) = J
2342. C IBBND(IBLL) = IBUB(J)
2343. C IZBND(IBLL) = IZBAR
2344. C
2345. C 200 IF (NTEMP(1) .LE. IBLB(J)) GO TO 500
2346. C ADD DOWN BRANCH ON BIJ TO LIST
2347. C IBLL = IBLL + 1
2348. C IBID(IBLL) = -J
2349. C IBBND(IBLL) = IBLB(J)
2350. C IZBND(IBLL) = IZBAR
2351. C
2352. C REVISE BOUNDS ON BIJ FOR FORWARD BRANCH
2353. C 500 IF (IBSTAR(K,J) .LT. IBUB(J)) IBUB(J) = IBSTAR(K,J)
2354. C IF (NTEMP(1) .GT. IBLB(J)) IBLB(J) = NTEMP(1)
2355. C
2356. C IF (NTEMP(1) .LT. IBLB(J)) GO TO 2000
2357. C MSTAT = 1
2358. C RETURN
2359. C
2360. C FORWARD BRANCH DOMINATED
2361. C 2000 MSTAT = 0
2362. C RETURN
2363. C END
2364. C SUBPOLTIME BBKTRK
2365. C
2366. C BACKTRACKS IN ALLOCATION-VARIABLES SEARCH TREE TO LOCATE NEXT
2367. C PROMISING NODE TO BE EXAMINED. LAST-IN-FIRST-OUT (LIFO)
2368. C SELECTION RULE USED.
2369. C
2370. C IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
2371. C 1 INTEGER*4 (I-N,Q)
2372. C INTEGER*2 IZSTAR,IBID,IBBND,IZBND
2373. C COMMON/BLOCK3/ RLAMDA(50,10),ZNAUT(10),ZJBAR(10),NSUBS,IZINC,
2374. C IZBAR,IDUM,IBLL,NJFIX,JFIXSM,IB,JFIX(10),NSOL(10),NLAMDA(10),
2375. C 2KSEG(10),IBSTAR(200,10),IBUB(10),IBLB(10),IBP(51,10),
2376. C 3IBUSED(200,10),IZSTAR(200,10),IBID(200),IBBND(200),IZBND(200)
2377. C RETURN TO MAIN IF LIST OF STORED NODES IS EMPTY (COMPUTATIONS
2378. C COMPLETED)

```

```

2379.      50 IF (IBLL .EQ. 0) RETURN
2380.      C      IDENTIFY BRANCHING VARIABLE BIJ CORRESPONDING TO LAST NODE ON
2381.      C      LIST
2382.      J = IBID(IBLL)
2383.      J = IABS(J)
2384.      C      TEST WHETHER SUBPROBLEM J IS CURRENTLY FIXED
2385.      IF (JFIX(J) .LE. 0) GO TO 100
2386.      C      "UNFIX" ALLOCATION TO SUBPROBLEM J NOW
2387.      NJFIX = NJFIX - 1
2388.      K = JFIX(J)
2389.      JFIXSM = JFIXSM - IZSTAR(K,J)
2390.      C
2391.      100 JFIX(J) = 0
2392.      IF (IZBND(IBLL) .LE. IZINC) GO TO 2000
2393.      C
2394.      C      GET NEXT NODE FROM LIST
2395.      IF (IBID(IBLL) .LT. 0) GO TO 200
2396.      C      NODE WAS CREATED BY UP-BRANCH ON BIJ
2397.      ITEMP = IBLB(J)
2398.      IBLB(J) = IBUB(J) + 1
2399.      IBUB(J) = IBBND(IBLL)
2400.      IZBAR = IZBND(IBLL)
2401.      JFIX(J) = -1
2402.      GO TO 1000
2403.      C
2404.      C      NODE WAS CREATED BY DOWN-BRANCH ON BIJ
2405.      200 ITEMP = IBUB(J)
2406.      IBUB(J) = IBLB(J) - 1
2407.      IBLB(J) = IBBND(IBLL)
2408.      IZBAR = IZBND(IBLL)
2409.      C
2410.      C      "BOOKKEEPING" TO ALLOW ORIGINAL BOUNDS ON BIJ TO BE RETRIEVED
2411.      C      LATER
2412.      1000 IBID(IBLL) = -IBID(IBLL)
2413.      IBBND(IBLL) = ITEMP
2414.      IZBND(IBLL) = -10000
2415.      RETURN
2416.      C
2417.      C      NODE FATHOMED: UPDATE BOUNDS AND BACKTRACK AGAIN
2418.      2000 IF (IBID(IBLL) .LT. 0) GO TO 2200
2419.      IBUB(J) = IBBND(IBLL)
2420.      GO TO 2500
2421.      C
2422.      2200 IBLB(J) = IBBND(IBLL)
2423.      C
2424.      2500 IBLL = IBLL - 1
2425.      GO TO 50
2426.      END
2427.      SUBROUTINE OUTSOL( INC)
2428.      C
2429.      C      OUTPUTS SOLUTION INFORMATION
2430.      C*****DESCRIPTION OF PARAMETERS*****
2431.      C      INC(J) = PARAMETER INDEXING THE SOLUTION IN SUBPROBLEM J
2432.      C      (J = 1,2,...,NSUBS) FROM WHICH THE OVERALL OPTIMUM IS
2433.      C      COMPRISED (INPUT)
2434.      C*****
2435.      C
2436.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
2437.      1      INTEGER*4 (I-N,Q)
2438.      INTEGER*2 IZSTAR,IBID,IBBND,IZBND
2439.      INTEGER*2 IAS,LAS

```



```

2440.      DIMENSION INC(10)
2441.      COMMON/BLOCK3/ RLAMDA(50,10),ZNAUT(10),ZJBAR(10),NSUBS,IZINC,
2442.      IIZBAR,IDOM,IBLL,NJFIX,JFIXSM,IB,JFIX(10),NSOL(10),NLAMDA(10),
2443.      2KSEG(10),IBSTAR(200,10),IBUB(10),IBLB(10),IBP(51,10),
2444.      3IBUSED(200,10),IZSTAR(200,10),IBID(200),IBBND(200),IZBND(200)
2445.      COMMON/BLOCK4/ XLBS(50,10),XUBS(50,10),AMATRS(1000),RHS(20,10),
2446.      INROWS(10),NCOLS(10),JFIRST(11),IAS(1000),LAS(50,10)
2447.      WRITE (6,10)
2448.      10 FORMAT ('1 OPTIMAL SOLUTION'/' SUBPROBLEM',T20,'SOL. NUMBER')
2449.      DO 100 J=1,NSUBS
2450.      K = INC(J)
2451.      100 WRITE (6,11) J,K
2452.      11 FORMAT (I7,T20,I5)
2453.      WRITE (6,12) IZINC
2454.      12 FORMAT ('/' MAXIMUM OBJECTIVE VALUE =',I10)
2455.      RETURN
2456.      END
2457.      SUBROUTINE PHASE1(J)
2458.      C
2459.      C      SOLVES SUBPROBLEM J AS A LINEAR PROGRAM PARAMETRICALLY IN THE
2460.      C      LINKING CONSTRAINT RIGHT-HAND SIDE. SIMULTANEOUSLY MODIFIES
2461.      C      THE RESULTING CONCAVE AND PIECEWISE-LINEAR PARAMETRIC OBJECTIVE
2462.      C      FUNCTION TO RENDER ALL BREAKPOINTS IBP(K,J) (K = 1,2,...,
2463.      C      NLAMDA(J)) INTEGRAL.
2464.      C
2465.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-h,Z), REAL*8 (B,D,X,Y),
2466.      1      INTEGER*4 (I-N,Q)
2467.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
2468.      INTEGER*2 IZSTAR,IBID,IBBND,IZBND
2469.      DOUBLE PRECISION E(1000)
2470.      DIMENSION A(500),INC(10)
2471.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
2472.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
2473.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
2474.      COMMON/BLOCK3/ RLAMDA(50,10),ZNAUT(10),ZJBAR(10),NSUBS,IZINC,
2475.      IIZBAR,IDOM,IBLL,NJFIX,JFIXSM,IB,JFIX(10),NSOL(10),NLAMDA(10),
2476.      2KSEG(10),IBSTAR(200,10),IBUB(10),IBLB(10),IBP(51,10),
2477.      3IBUSED(200,10),IZSTAR(200,10),IBID(200),IBBND(200),IZBND(200)
2478.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
2479.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
2480.      2      IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
2481.      3      JCULP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
2482.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
2483.      C      GET OPTIMAL LP SOLUTION WITH RHS = IBLB(J)
2484.      ITSINV = 99999
2485.      CALL NCRML
2486.      ZNAUT(J) = SNGL(X(IOBJ))
2487.      IBBAR = IBLB(J)
2488.      K = 0
2489.      C
2490.      25 K = K + 1
2491.      IBP(K,J) = IBBAR
2492.      C
2493.      C      COMPUTE NEXT BREAKPOINT VALUE BBAR
2494.      CALL UNPACK(2)
2495.      CALL FTRAN(1)
2496.      IF (Y(IOBJ) .LE. ZTCOST) GO TO 1000
2497.      DP = 1.E10
2498.      DO 100 I=1,NROW
2499.      IF (I .EQ. IOBJ) GO TO 100
2500.      JCULP = JH(I)

```



```

2501.      IF (Y(I) .GE. ZTOLPV) GO TO 60
2502.      IF (Y(I) .LE. -ZTOLPV) GO TO 50
2503.      GO TO 100
2504.      50 DY = XLB(JCGLP)
2505.      GO TO 70
2506.      60 DY = XUB(JCOLP)
2507.      70 DE = (CY - X(I))/Y(I)
2508.      IF (DP .LE. DE) GO TO 100
2509.      DP = DE
2510.      IROWP = I
2511.      100 CONTINUE
2512.      DTEMP = FLOAT(IBBAR) + DP
2513.      DE = DTEMP + ZTOLZE
2514.      IBBAR = IDINT(DE)
2515.      IF (DE .LT. 0.) IBBAR = IBBAR - 1
2516.      IF (IBBAR .GE. IBUB(J)) GO TO 1000
2517.      TEMP = FLOAT(IBBAR)
2518.      DE = DTEMP - TEMP
2519.      IF (IBBAR .EQ. IBP(K,J)) GO TO 500
2520.      IF (DE .LT. ZTOLZE) GO TO 400
2521.      GO TO 600
2522.      C
2523.      C      NEW BREAKPCINT BBAR NATURALLY INTEGER
2524.      400 RLAMDA(K,J) = SNGL(Y(IOBJ))
2525.      C      UPDATE X TO REFLECT VALUES AT NEW BBAR, THEN PIVOT
2526.      DO 41C I=1,NROW
2527.      41C X(I) = X(I) + Y(I)*DP
2528.      B(2) = FLOAT(IBBAR)
2529.      CALL NORMAL
2530.      GO TO 25
2531.      C
2532.      C      NEW BREAKPCINT BBAR < PREVIOUS BBAR + 1
2533.      500 DTEMP = X(IOBJ)
2534.      IBBAR = IBBAR + 1
2535.      C      UPDATE X:  NEW BBAR = OLD BBAR + 1
2536.      DO 51C I=1,NROW
2537.      51C X(I) = X(I) + Y(I)
2538.      B(2) = FLOAT(IBBAR)
2539.      CALL NORMAL
2540.      RLAMDA(K,J) = SNGL(X(IOBJ) - DTEMP)
2541.      GO TO 25
2542.      C
2543.      C      NEW BREAKPCINT BBAR > OLD BBAR + 1, BUT NON-INTEGER
2544.      600 RLAMDA(K,J) = SNGL(Y(IOBJ))
2545.      K = K + 1
2546.      IBP(K,J) = IBBAR
2547.      DTEMP = X(IOBJ) + Y(IOBJ)*(DP - DE)
2548.      IBBAR = IBBAR + 1
2549.      DP = DP + (1. - DE)
2550.      DO 61C I=1,NROW
2551.      61C X(I) = X(I) + Y(I)*DP
2552.      B(2) = FLOAT(IBBAR)
2553.      CALL NORMAL
2554.      RLAMDA(K,J) = SNGL(X(IOBJ) - DTEMP)
2555.      GO TO 25
2556.      C
2557.      C      END OF PLP
2558.      1000 RLAMDA(K,J) = SNGL(Y(IOBJ))
2559.      NLAMDA(J) = K
2560.      IBP(K+1,J) = IBUB(J)
2561.      RETURN

```

2562.

END

\*\*\*\*\*SAMPLE INPUT DATA FOR DECOMPOSITION CODE DSLC\*\*\*\*\*

```

      4      293      1      0
1
NAME      SUB1
RCWS
N OBJ
L RCW1
L RCW11
L ROW12
L ROW13
L RCW14
COLUMNS
COL11      CBJ      -56.      ROW1      40.
COL11      RCW11     16.      ROW12     38.
COL11      RCW13      8.      ROW14     38.
COL12      CBJ     -113.      ROW1      91.
COL12      RCW11     92.      ROW12     39.
COL12      RCW13     71.      ROW14     52.
COL13      CBJ     -30.      ROW1      10.
COL13      RCW11     41.      ROW12     32.
COL13      RCW13     30.      ROW14     30.
COL14      CBJ     -62.      ROW1      30.
COL14      RCW11     16.      ROW12     71.
COL14      RCW13     60.      ROW14     42.
COL15      CBJ    -210.      ROW1     160.
COL15      RCW11    150.      ROW12     80.
COL15      RCW13    200.      ROW14    170.
RHS
RHS1      RCW11     221.      ROW12    182.
RHS1      RCW13     258.      ROW14    232.
ENDATA
0.  0.  0.  0.  C.
1.  1.  1.  1.  1.
      C      200
2
NAME      SUB2
RCWS
N OBJ
L RCW1
L RCW21
L ROW22
L ROW23
L RCW24
COLUMNS
COL21      CBJ     -43.      ROW1      20.
COL21      RCW21     23.      ROW22     26.
COL21      RCW23     18.      ROW24      9.
COL22      CBJ      -7.      ROW1      3.
COL22      RCW21      4.      ROW22      5.
COL22      RCW23      6.      ROW24      7.
COL23      CBJ    -33.      ROW1     12.
COL23      RCW21     18.      ROW22     40.
COL23      RCW23     30.      ROW24     20.
COL24      CBJ      -5.      ROW1      3.
COL24      RCW21      6.      ROW22      8.
COL24      RCW23      4.
COL25      CBJ    -12.      ROW1     18.

```



COL25	RCW22	12.	ROW23	8.
CCL25	RCW24	3.		
RHS				
RHS1	RCW21	36.	ROW22	64.
RHS1	RCW23	46.	ROW24	27.

ENDATA

0.	0.	0.	0.	0.
1.	1.	1.	1.	1.
	0		44	

3  
NAME SUB3  
RCWS

N OBJ  
L RCW1  
L RCW31  
L RCW32  
L RCW33  
L RCW34

COLUMNS				
COL31	CBJ	-32.	ROW1	9.
COL31	RCW31	12.	ROW32	30.
CCL31	RCW33	31.	ROW34	21.
COL32	CBJ	-20.	ROW1	25.
COL32	RCW31	8.	ROW32	15.
COL32	RCW33	6.	ROW34	4.
COL33	CBJ	-4.	ROW1	1.
CCL33	RCW31	2.	ROW33	3.
COL33	RCW34	1.		
COL34	CBJ	-3.	ROW1	1.
COL34	RCW31	1.	ROW32	1.
COL34	RCW34	2.		
CCL35	CBJ	-43.	ROW1	10.
COL35	RCW32	23.	ROW33	18.
CCL35	RCW34	14.		

RHS				
RHS1	RCW31	16.	ROW32	48.
RHS1	RCW33	41.	ROW34	29.

ENDATA

0.	0.	0.	0.	0.
1.	1.	1.	1.	1.
	0		37	

4  
NAME SUB4  
RCWS

N OBJ  
L ROW1  
L RCW41  
L RCW42  
L RCW43  
L RCW44

COLUMNS				
COL41	CBJ	-426.	ROW1	280.
COL41	RCW41	200.	ROW42	100.
COL41	RCW43	60.	ROW44	310.
COL42	CBJ	-42.	ROW1	10.
CCL42	RCW41	20.	ROW43	21.
COL42	RCW44	8.		
COL43	CBJ	-12.	ROW1	8.
COL43	RCW41	6.	ROW42	20.
CCL43	RCW43	4.	ROW44	4.
COL44	CBJ	-8.	ROW1	1.

CGL44	RCW41	2.	ROW42	3.
CGL44	RCW44	6.		
CGL45	C8J	-2.	ROW1	1.
CGL45	RCW41	1.	ROW43	2.
CGL45	RCW44	1.		
RHS				
RHS1	RCW41	160.	ROW42	86.
RHS1	RCW43	61.	ROW44	230.
ENDATA				
0.	0.	0.	0.	0.
1.	1.	1.	1.	1.
	0			
		20		



\*\*\*\*\*OUTPUT GENERATED BY DSLC ON SAMPLE PROBLEM\*\*\*\*\*

START NEW PROBLEM

IBP(K, 1) = 0 10 11 40 41 80 81 162 163 185 186 193 194 200  
 RLAMDA(K, 1) = 3.00 2.07 2.07 1.40 1.40 1.31 1.31 1.22 1.22 1.18 1.03 1.03 1.01  
 ZNAUT( 1) = -0.0000

IBP(K, 2) = 0 12 13 15 16 27 28 36 37 39 40 42 43 44 44  
 RLAMDA(K, 2) = 2.75 2.33 2.33 2.11 2.11 1.60 0.64 0.63 0.62 0.62 0.61 0.59 0.37 0.2  
 ZNAUT( 2) = -0.0000

IBP(K, 3) = 0 10 11 12 16 17 18 30 31 34 35 37  
 RLAMDA(K, 3) = 4.30 4.00 3.56 3.56 3.45 0.67 0.60 0.59 0.57 0.32 0.20  
 ZNAUT( 3) = -0.0000

IBP(K, 4) = 0 1 2 11 12 13 20  
 RLAMDA(K, 4) = 8.00 4.20 4.20 2.00 1.52 1.52  
 ZNAUT( 4) = -0.0000

SUB 1 SOL 1: 266 0 55 64 50 24 1 0 0 0 1  
 SUB 2 SOL 1: 62 1 9 21 14 8 1 1 0 0 1  
 SUB 3 SOL 1: 50 20 13 24 20 12 0 0 1 1 1  
 SUB 4 SOL 1: 64 0 131 63 34 211 0 1 1 1 1  
 TIME = 1.10 SECONDS: IZINC = 442  
 SUB 4 SOL 2: 62 0 132 63 36 212 0 1 1 1 0  
 SUB 3 SOL 1: 50 21 0 0 0 0 0 0 0 0 0  
 SUB 3 SOL 1: 50 22 0 0 0 0 0 0 0 0 0  
 SUB 4 SOL 3: 52 5 137 83 38 215 0 1 0 1 1  
 SUB 3 SOL 2: 70 0 5 9 14 8 0 1 1 1 1  
 TIME = 1.62 SECONDS: IZINC = 450  
 SUB 2 SOL 2: 55 2 13 26 20 15 1 0 0 0 1  
 SUB 3 SOL 3: 63 0 8 10 17 11 0 1 0 0 1  
 SUB 3 SOL 4: 67 0 6 10 14 10 0 1 1 0 1  
 SUB 4 SOL 4: 54 0 134 66 36 218 0 1 1 0 0  
 SUB 2 SOL 3: 55 11 3 25 18 11 1 1 0 1 0  
 TIME = 2.26 SECONDS: IZINC = 452  
 TIME = 2.29 SECONDS: IZINC = 455  
 SUB 2 SOL 4: 50 2 9 33 22 11 1 1 0 0 0  
 SUB 2 SOL 5: 67 0 3 13 10 8 1 1 0 1 1  
 SUB 2 SOL 1: 62 2 0 0 0 0 0 0 0 0 0  
 SUB 1 SOL 2: 261 28 56 2 89 70 1 1 1 1 0  
 TIME = 2.92 SECONDS: IZINC = 457  
 SUB 1 SOL 3: 240 0 30 70 28 32 0 0 1 0 1  
 TIME = 3.10 SECONDS: IZINC = 462

OPTIMAL SOLUTION



SUBPROBLEM	SOL. NUMBER
1	2
2	5
3	2
4	1

MAXIMUM OBJECTIVE VALUE = 462

TOTAL SOLUTION TIME = 3.12 SECONDS

TIME IN PHASE I = 0.60 SECONDS

MASTER PROBLEM SOLVED 91 TIMES: TOTAL TIME SOLVING MASTER PROBLEM = 0.22 SECONDS

NUMBER OF TIMES EACH SUBPROBLEM SOLVED = 3 6 6 4

NUMBER OF DISTINCT SOLUTIONS OBTAINED IN EACH SUBPROBLEM = 3 5 4 4

TOTAL TIME SOLVING SUBPROBLEMS = 2.20 SECONDS

### APPENDIX III

LISTING AND SAMPLE INPUT/OUTPUT FOR COMPUTER PROGRAM DMLC

\*\*\*\*\*OUTPUT GENERATED BY DMLC ON SAMPLE PROBLEM\*\*\*\*\*

```

SLP 1 SCL 1: 266 0 55 64 50 24 1 0 0 0 1
SLP 2 SCL 1: 62 1 9 21 14 8 1 1 0 0 1
SLP 3 SCL 1: 50 20 13 24 20 12 0 0 1 1 1
SLP 4 SCL 1: 64 0 131 63 34 211 0 1 1 1 1
TIME = 1.64 SECONDS: IZINC = 442
SLP 3 SCL 2: 47 0 14 25 20 14 0 0 1 0 1
SLP 3 SCL 3: 50 0 0 0 0 0 0 0 0 0 0
SLP 3 SCL 4: 50 0 0 0 0 0 0 0 0 0 0
SLP 3 SCL 5: 63 0 8 10 17 11 0 1 0 0 1
SLP 4 SCL 2: 52 5 137 83 38 215 0 1 0 1 1
TIME = 2.44 SECONDS: IZINC = 443
SLP 3 SCL 6: 70 0 5 9 14 8 0 1 1 1 1
SLP 4 SCL 3: 50 0 138 83 40 216 0 1 0 1 0
TIME = 2.65 SECONDS: IZINC = 448
SLP 3 SCL 7: 67 0 6 10 14 10 0 1 1 0 1
SLP 2 SCL 2: 55 2 13 26 20 15 1 0 0 0 1
SLP 2 SCL 3: 55 11 3 25 18 11 1 1 0 1 0
SLP 4 SCL 4: 62 0 132 63 36 212 0 1 1 1 0
TIME = 4.19 SECONDS: IZINC = 450
SLP 4 SCL 5: 54 0 134 66 36 218 0 1 1 0 0
TIME = 4.45 SECONDS: IZINC = 455
SLP 2 SCL 4: 50 2 9 33 22 11 1 1 0 0 0
SLP 2 SCL 5: 67 0 3 13 10 8 1 1 0 1 1
SLP 2 SCL 6: 62 0 0 0 0 0 0 0 0 0 0
SLP 1 SCL 2: 261 28 56 2 89 70 1 1 1 1 0
TIME = 7.05 SECONDS: IZINC = 457
SLP 1 SCL 3: 240 0 30 70 28 32 0 0 1 0 1
SLP 1 SCL 4: 231 8 57 34 119 100 1 1 0 1 0
TIME = 10.25 SECONDS: IZINC = 462
OPTIMAL SOLUTION

```

```

SLP/PROBLEM      SCL. NUMBER
1                  2
2                  5
3                  6
4                  1

```

MAXIMUM OBJECTIVE VALUE = 462

TOTAL SOLUTION TIME = 10.96SECONDS

TIME IN PHASE1 = 0.85 SECONDS

MASTER PROBLEM SOLVED 229 TIMES: TOTAL TIME SOLVING MASTER PROBLEM = 7.62 SECONDS

NUMBER OF TIMES EACH SUBPROBLEM SOLVED = 4 6 7 5

NUMBER OF DISTINCT SOLUTIONS OBTAINED IN EACH SUBPROBLEM = 4 6 7 5

TOTAL TIME SOLVING SUBPROBLEMS = 2.35 SECONDS

40/0



CCL42	CBJ	-42.	ROW1	10.
CCL42	RCW41	20.	ROW43	21.
CCL42	RCW44	0.		
CCL43	CBJ	-12.	ROW1	8.
CCL43	RCW41	0.	ROW42	20.
CCL43	RCW43	4.	ROW44	4.
CCL44	CBJ	-3.	ROW1	1.
CCL44	RCW41	2.	ROW42	3.
CCL44	RCW44	0.		
CCL45	CBJ	-2.	ROW1	1.
CCL45	RCW41	1.	ROW43	2.
CCL45	RCW44	1.		
RFS				
RFS1	RCW1	10.		
RFS1	RCW41	100.	ROW42	86.
RFS1	RCW43	01.	ROW44	230.
ENDATA				
C.	C.	C.	O.	C.
1.	1.	1.	1.	1.
	C			
	20			

CCL24	FCW21	6.	ROW22	6.
CCL24	FCW23	4.		
CCL25	CEJ	-12.	ROW1	18.
CCL25	FCW22	12.	ROW23	8.
CCL25	FCW24	3.		
RFS				
RFS1	FCW1	22.		
RFS1	FCW21	50.	ROW22	64.
RFS1	FCW23	46.	ROW24	27.

# ENCATA

C.	C.	C.	C.	C.
1.	1.	1.	1.	1.
	C			
	44			

3  
NAME SLE3  
ROWS

N OBJ  
L RCW1  
L RCW31  
L RCW32  
L RCW33  
L RCW34

# COLUMNS

CCL31	CEJ	-32.	ROW1	9.
CCL31	FCW31	12.	ROW32	30.
CCL31	FCW33	31.	ROW34	21.
CCL32	CEJ	-20.	ROW1	25.
CCL32	FCW31	8.	ROW32	15.
CCL32	FCW33	0.	ROW34	4.
CCL33	CEJ	-4.	ROW1	1.
CCL33	FCW31	2.	ROW33	3.
CCL33	FCW34	1.		
CCL34	CEJ	-3.	ROW1	1.
CCL34	FCW31	1.	ROW32	1.
CCL34	FCW34	2.		
CCL35	CEJ	-43.	ROW1	10.
CCL35	FCW32	23.	ROW33	18.
CCL35	FCW34	14.		

RFS				
RFS1	FCW1	18.		
RFS1	FCW31	10.	ROW32	48.
RFS1	FCW33	41.	ROW34	29.

# ENCATA

C.	C.	C.	C.	C.
1.	1.	1.	1.	1.
	C			
	37			

4  
NAME SLE4  
ROWS

N OBJ  
L RCW1  
L RCW41  
L RCW42  
L RCW43  
L RCW44

# COLUMNS

CCL41	CEJ	-420.	ROW1	280.
CCL41	FCW41	200.	ROW42	100.
CCL41	FCW43	60.	ROW44	310.

\*\*\*\*\*SAMPLE INFLT DATA FOR DECOMPOSITION CODE DMLC\*\*\*\*\*

4 1 1 0  
292

1  
NAME SLB1  
ROWS  
N OBJ  
L ROW1  
L ROW11  
L ROW12  
L ROW13  
L ROW14  
COLUMNS  
CCL11 CEJ -50. ROW1 40.  
CCL11 FCW11 10. ROW12 38.  
CCL11 FCW13 5. ROW14 38.  
CCL12 CEJ -113. ROW1 91.  
CCL12 FCW11 92. ROW12 39.  
CCL12 FCW13 71. ROW14 52.  
CCL13 CEJ -30. ROW1 10.  
CCL13 FCW11 41. ROW12 32.  
CCL13 FCW13 30. ROW14 30.  
CCL14 CEJ -62. ROW1 30.  
CCL14 FCW11 16. ROW12 71.  
CCL14 FCW13 60. ROW14 42.  
CCL15 CEJ -210. ROW1 160.  
CCL15 FCW11 150. ROW12 80.  
CCL15 FCW13 200. ROW14 170.  
RFS  
RFS1 FCW1 100.  
RFS1 FCW11 221. ROW12 182.  
RFS1 FCW13 298. ROW14 232.

ENDATA  
C. C. C. C. C.  
1. 1. 1. 1. 1.  
C  
200

2  
NAME SLB2  
ROWS  
N OBJ  
L ROW1  
L ROW21  
L ROW22  
L ROW23  
L ROW24  
COLUMNS  
CCL21 CEJ -43. ROW1 20.  
CCL21 FCW21 23. ROW22 26.  
CCL21 FCW23 18. ROW24 9.  
CCL22 CEJ -7. ROW1 3.  
CCL22 FCW21 4. ROW22 5.  
CCL22 FCW23 0. ROW24 7.  
CCL23 CEJ -33. ROW1 12.  
CCL23 FCW21 18. ROW22 40.  
CCL23 FCW23 30. ROW24 20.  
CCL24 CEJ -5. ROW1 3.



```

2684.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
2685.      1      INTEGER*4 (I-N,L)
2686.      INTEGER*2 IZSTAR,IBID,IBBND,IZBND
2687.      INTEGER*2 IAS,LAS
2688.      DIMENSION INC(10)
2689.      COMMON/BLOCK3/ KLAMCA(5,20,10),ZNAUT(20,10),ZJBAR(10),NLINK,NSUBS,
2690.      1 IZINC,IZBAR,IBLM,IBLL,NJFIX,JFIXSM,IB(5),JFIX(10),NSOL(10),
2691.      2 NLAMDA(10),IBSTAR(5,40,10),IBUSED(5,40,10),IEUB(5,10),IBLB(5,10),
2692.      3 IZSTAR(40,10),IBID(2,200),IBBND(200),IZBND(200)
2693.      COMMON/BLOCK4/ XLBS(20,10),XUBS(20,10),AMATRS(1000),RHS(10,10),
2694.      INROWS(10),NCOLS(10),JFIRST(11),IAS(1000),LAS(20,10)
2695.      WRITE (6,10)
2696.      10 FORMAT ('1 OPTIMAL SOLUTION'/' SUBPROBLEM',T20,'SOL. NUMBER')
2697.      DO 100 J=1,NSUBS
2698.      K = INC(J)
2699.      100 WRITE (6,11) J,K
2700.      11 FORMAT (17,T20,15)
2701.      WRITE (6,12) IZINC
2702.      12 FORMAT (17,' MAXIMUM OBJECTIVE VALUE =',110)
2703.      RETURN
2704.      END

```

```

2623.      3IZSTAR(40,10),IBID(2,200),IBBND(200),IZBND(200)
2624.      C      RETURN TO MAIN IF LIST OF STORED NODES IS EMPTY (COMPUTATIONS
2625.      C      COMPLETED)
2626.      50 IF (IBLL .EQ. 0) RETURN
2627.      C      IDENTIFY BRANCHING VARIABLE BIJ CORRESPONDING TO LAST NODE ON
2628.      C      LIST
2629.      I = IBID(1,IBLL)
2630.      J = IBID(2,IBLL)
2631.      C      TEST WHETHER SUBPROBLEM J IS CURRENTLY FIXED
2632.      IF (JFIX(J) .EQ. 0) GO TO 100
2633.      C      "UNFIX" ALLOCATION TO SUBPROBLEM J NOW
2634.      NJFIX = NJFIX - 1
2635.      K = JFIX(J)
2636.      JFIXSM = JFIXSM - IZSTAR(K,J)
2637.      JFIX(J) = 0
2638.      C
2639.      100 IF (IZBND(IBLL) .LE. IZINC) GO TO 2000
2640.      C
2641.      C      GET NEXT NODE FROM LIST
2642.      IF (I .LT. 0) GO TO 200
2643.      C      NODE WAS CREATED BY UP-BRANCH ON BIJ
2644.      ITEMP = IBLB(I,J)
2645.      IBLB(I,J) = IBUB(I,J) + 1
2646.      IBUB(I,J) = IBEND(IBLL)
2647.      IZBAR = IZBND(IBLL)
2648.      GO TO 1000
2649.      C
2650.      C      NODE WAS CREATED BY DOWN-BRANCH ON BIJ
2651.      200 I = -I
2652.      ITEMP = IBUB(I,J)
2653.      IBLB(I,J) = IBLB(I,J) - 1
2654.      IBLB(I,J) = IBEND(IBLL)
2655.      IZBAR = IZBND(IBLL)
2656.      C
2657.      C      "BOOKKEEPING" TO ALLOW ORIGINAL BOUNDS ON BIJ TO BE RETRIEVED
2658.      C      LATER
2659.      1000 IBID(1,IBLL) = -IBID(1,IBLL)
2660.      IBBND(IBLL) = ITEMP
2661.      IZBND(IBLL) = -10000
2662.      RETURN
2663.      C
2664.      C      NODE FATHOMED: UPDATE BOUNDS AND BACKTRACK AGAIN
2665.      2000 IF (I .LT. 0) GO TO 2200
2666.      IBLB(I,J) = IBEND(IBLL)
2667.      GO TO 2500
2668.      C
2669.      2200 I = -I
2670.      IBLB(I,J) = IBEND(IBLL)
2671.      C
2672.      2500 IBLL = IBLL - 1
2673.      GO TO 50
2674.      END
2675.      SUBROUTINE CUTSOL(INC)
2676.      C
2677.      C      CUTSOL: SOLUTION INFORMATION
2678.      C*****DESCRIPTION OF PARAMETERS*****
2679.      C      INC(J) = PARAMETER INDEXING THE SOLUTION IN SUBPROBLEM J
2680.      C      (J = 1,2,...,NSUBS) FROM WHICH THE OVERALL OPTIMUM IS
2681.      C      COMPRISED (INPUT)
2682.      C*****
2683.      C

```

```

2562. C
2563. IMPLICIT REAL*4 (A,C,E-H,C,P,R-W,Z), REAL*8 (B,D,X,Y),
2564. 1 INTEGER*4 (I-N,Q)
2565. INTEGER*2 JH,KINBAS,LA,LE,IA,IE
2566. INTEGER*2 IZSTAR,IBID,IBBND,IZBND
2567. DOUBLE PRECISION E(1000)
2568. REAL A(500)
2569. COMMON/BLCK3/ KLAMDA(5,20,10),ZNAUT(20,10),ZJBAR(10),NLINK,NSUBS,
2570. IZINC,IZBAR,IDCM,IBLL,NJFIX,JFIXSM,IB(5),JFIX(10),NSQL(10),
2571. 2NLAMDA(10),IBSTAR(5,40,10),IBUSED(5,40,10),IBUB(5,10),IBLB(5,10),
2572. 3IZSTAR(40,10),IBID(2,200),IBBND(200),IZBND(200)
2573. COMMON E,XLE(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
2574. 1 DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
2575. 2 IUBJ,IRCWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
2576. 3 JGCLP,NRUA,NCLL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
2577. 4 IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
2578. DO 1000 I=1,NLINK
2579. NTEMP(I) = IBUSED(I,K,J)
2580. IF (IBSTAR(I,K,J) .GE. IBUB(I,J)) GO TO 200
2581. C ADD UP BRANCH ON BIJ TO LIST
2582. IBLL = IBLL + 1
2583. IBID(1,IBLL) = I
2584. IBID(2,IBLL) = J
2585. IBBND(IBLL) = IBUB(I,J)
2586. IZBND(IBLL) = IZBAR
2587. C
2588. 200 IF (NTEMP(I) .LE. IBLB(I,J)) GO TO 500
2589. C ADD DOWN BRANCH ON BIJ TO LIST
2590. IBLL = IBLL + 1
2591. IBID(1,IBLL) = -I
2592. IBID(2,IBLL) = J
2593. IBBND(IBLL) = IBLB(I,J)
2594. IZBND(IBLL) = IZBAR
2595. C
2596. C REVISE BOUNDS ON BIJ FOR FORWARD BRANCH
2597. 500 IF (IBSTAR(I,K,J) .LT. IBUB(I,J)) IBUB(I,J) = IBSTAR(I,K,J)
2598. IF (NTEMP(I) .LT. IBLB(I,J)) IBLB(I,J) = NTEMP(I)
2599. 1000 CONTINUE
2600. C
2601. DO 1500 I=1,NLINK
2602. IF (NTEMP(I) .LT. IBLB(I,J)) GO TO 2000
2603. 1500 CONTINUE
2604. MSTAT = 1
2605. RETURN
2606. C
2607. C FORWARD BRANCH DOMINATED
2608. 2000 MSTAT = 0
2609. RETURN
2610. END
2611. SUBROUTINE BBKTK
2612. C
2613. C BACKTRACKS IN ALLOCATION-VARIABLES SEARCH TREE TO LOCATE NEXT
2614. C PRECISING NODE TO BE EXAMINED. LAST-IN-FIRST-OUT (LIFO)
2615. C SELECTION RULE USED.
2616. C
2617. IMPLICIT REAL*4 (A,C,E-H,C,P,R-W,Z), REAL*8 (B,D,X,Y),
2618. 1 INTEGER*4 (I-N,Q)
2619. INTEGER*2 IZSTAR,IBID,IBBND,IZBND
2620. COMMON/BLCK3/ KLAMDA(5,20,10),ZNAUT(20,10),ZJBAR(10),NLINK,NSUBS,
2621. IZINC,IZBAR,IDCM,IBLL,NJFIX,JFIXSM,IB(5),JFIX(10),NSQL(10),
2622. 2NLAMDA(10),IBSTAR(5,40,10),IBUSED(5,40,10),IBUB(5,10),IBLB(5,10),

```



```

2501.      DOUBLE PRECISION E(1000)
2502.      REAL A(500)
2503.      COMMON/BLOCK2/ DEPART(122),KEVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
2504.      1      NP1VCT,IPTYPE,RCCST,IFEAS,IPART(122),INCUMB(122),
2505.      2      IVBND(500),IVID(500),IOBND(500)
2506.      COMMON/BLOCK3/ KLAMDA(5,20,10),ZNAUT(20,10),ZJBAR(10),NLINK,NSUBS,
2507.      1IZINC,IZBAR,IDUM,IBLL,NJFIX,JFIXSM,IB(5),JFIX(10),NSOL(10),
2508.      2NLAMDA(10),IBSTAR(5,40,10),IBUSED(5,40,10),IBUB(5,10),IBLB(5,10),
2509.      3IZSTAR(40,10),IBID(2,200),IBBND(200),IZBND(200)
2510.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,OPRUD,
2511.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
2512.      2      IUBJ,IRKWP,IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,ITSINV,IFFEZ,
2513.      3      JCULP,NKRW,NCUL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
2514.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
2515.      C      TEST WHETHER SUBPROBLEM J WAS FEASIBLE (IFEAS = 1)
2516.      IF (IFEAS.EQ. 1) GO TO 90
2517.      NP1 = NLINK + 1
2518.      INCVAL = -10000
2519.      INCUMB(1) = -10000
2520.      DO 10 L = 2,NP1
2521.      10 INCUMB(L) = 10000
2522.      NP1 = NP1 + 1
2523.      DO 20 L = NP1,NCOL
2524.      20 INCUMB(L) = -1
2525.      C
2526.      50 LL = NSOL(J)
2527.      NEW = LL + 1
2528.      DO 95 I=1,NLINK
2529.      95 IBUSEC(I,NEW,J) = IBSTAR(I,NEW,J) - INCUMB(I+1)
2530.      IF (LL.EQ. 0) GO TO 1000
2531.      C      COMPARE NEW SOLUTION IN SUB J WITH PREVIOUS SOLUTIONS
2532.      DO 500 K=1,LL
2533.      DO 200 L=1,NLINK
2534.      IF (IBUSED(L,NEW,J).NE. IBUSED(L,K,J)) GO TO 500
2535.      200 CONTINUE
2536.      C      NEW SOLUTION SAME AS KTH: CHECK IF NEW REGION INCLUDES KTH
2537.      DO 300 I=1,NLINK
2538.      IF (IBSTAR(I,K,J) - IBSTAR(I,NEW,J)) 300,300,500
2539.      300 CONTINUE
2540.      C      NEW REGION INCLUDES KTH: REWRITE OVER KTH REGION
2541.      DO 400 I=1,NLINK
2542.      400 IBSTAR(I,K,J) = IBSTAR(I,NEW,J)
2543.      NTEMP(2) = K
2544.      RETURN
2545.      500 CONTINUE
2546.      C
2547.      C      NEW SOLUTION DIFFERS FROM ALL PREVIOUS SOLUTIONS
2548.      1000 NSOL(J) = NEW
2549.      IZSTAR(NEW,J) = INCUMB(1)
2550.      NTEMP(2) = NEW
2551.      RETURN
2552.      END
2553.      SUBROUTINE LMBRAN(J,K)
2554.      C
2555.      C      BRANCHES ON ALLOCATION VARIABLES BIJ IN SUBPROBLEM J
2556.      C*****DESCRIPTION OF PARAMETERS*****
2557.      C      J = PARAMETER INDEXING SUBPROBLEM IN WHICH BRANCHING IS DONE
2558.      C      K = PARAMETER INDEXING A PREVIOUSLY EXAMINED ALLOCATION IN
2559.      C      SUBPROBLEM J. THE CURRENT ALLOCATION IN SUBPROBLEM J IS TO
2560.      C      BE FIXED AT THE KTH LEVEL FOLLOWING THE BRANCHING PROCEDURE
2561.      C*****

```

```

2440.      DO 210 K=LL,NCLL
2441.      210 INCLMB(K) = 0
2442.      IF EAS = 1
2443.      RETURN
2444.      END
2445.      SUBROUTINE PISAVR(J)
2446.      C
2447.      C      CHECKS IF LP-OPTIMAL DUAL MULTIPLIERS JUST OBTAINED IN
2448.      C      SUBPROBLEM J ARE DISTINCT FROM PREVIOUS SETS OF MULTIPLIERS.
2449.      C      IF DISTINCT, STORES NEW MULTIPLIERS ASSOCIATED WITH LINKING
2450.      C      CONSTRAINTS.
2451.      C
2452.      IMPLICIT REAL*4 (A,C,E-H,G,P,R-W,Z), REAL*8 (B,D,X,Y),
2453.      1      INTEGER*4 (I-N,Q)
2454.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
2455.      INTEGER*2 IZSTAR,IBID,IBBND,IZBND
2456.      DOUBLE PRECISION E(1000)
2457.      REAL A(500)
2458.      COMMON/BLCK/ ZTOLZE,ZTOLPV,ZTCCST,ZTOLRJ,ZTCLSM,CRO,QMA,QBA,QFI,
2459.      1      CEG,CBL,QA,QPL,QMI,CZ,QI,QF,CN,CU,NRMAX,NEMAX,CB,QC,
2460.      2      QE,QF,QL,QU,QR,QM,QG,NTMAX
2461.      COMMON/BLCK3/ RLAMDA(5,20,10),ZNAUT(20,10),ZJBAR(10),NLINK,NSUBS,
2462.      1      IZINC,IZBAR,IDCM,IBLL,NJFIX,JFIXSM,IB(5),JFIX(10),NSOL(10),
2463.      2      ZNLAMDA(10),IBSTAR(5,40,10),IBUSED(5,40,10),IBUB(5,10),IBLB(5,10),
2464.      3      IZSTAR(40,10),IBID(2,200),IBBND(200),IZBND(200)
2465.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
2466.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
2467.      2      IUBJ,IRCP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
2468.      3      JCULP,NROW,NCLL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
2469.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
2470.      LL = NLAMDA(J)
2471.      IF (LL.EQ. 0) GO TO 1000
2472.      C      TEST IF OPTIMAL DUAL MULTIPLIERS DISTINCT FROM PREVIOUS SETS
2473.      C      DO 100 K=1,LL
2474.      C      DO 50 I=1,NLINK
2475.      C      IF (DABS(Y(I+1) - RLAMDA(I,K,J)) .GT. ZTOLZE) GO TO 100
2476.      50 CONTINUE
2477.      RETURN
2478.      100 CONTINUE
2479.      C
2480.      C      NEW MULTIPLIERS DISTINCT; STORE THEM
2481.      1000 LL = LL + 1
2482.      C      DO 1150 I=1,NLINK
2483.      1150 RLAMDA(I,LL,J) = SNGL(Y(I+1))
2484.      DP = X(IUBJ)
2485.      K = NLINK + 1
2486.      DO 1200 I=2,K
2487.      1200 DP = DP - Y(I)*B(I)
2488.      ZNAUT(LL,J) = SNGL(DP)
2489.      NLAMDA(J) = LL
2490.      RETURN
2491.      END
2492.      SUBROUTINE SAVER(J)
2493.      C
2494.      C      STORES ALLOCATION AND SOLUTION INFORMATION FOR SUBPROBLEM J
2495.      C
2496.      IMPLICIT REAL*4 (A,C,E-H,G,P,R-W,Z), REAL*8 (B,D,X,Y),
2497.      1      INTEGER*4 (I-N,Q)
2498.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
2499.      INTEGER*2 IPART,INCLMB,IVBND,IVID,ICBND
2500.      INTEGER*2 IZSTAR,IBID,IBBND,IZBND

```



```

2379.      LL = NROW - K
2380.      DO 40 I=1,LL
2381.      40 B(K+1) = USLE(KHST(I,J))
2382.      DO 50 I=1,NROW
2383.      JH(I) = 1
2384.      50 KINBAS(I) = 1
2385.      LL = NROW + 1
2386.      DO 60 I=LL,NCOL
2387.      60 KINBAS(I) = 0
2388.      ITCNT = 0
2389.      INVERQ = NROW
2390.      RETURN
2391.      END
2392.      SUBROUTINE GETLBD(J)
2393.      C
2394.      C      COMPARES CURRENT ALLOCATION IN SUBPROBLEM J WITH PREVIOUSLY
2395.      C      EXAMINED ALLOCATIONS IN ORDER TO DETERMINE AN INITIAL LOWER
2396.      C      BOUND ON THE MAXIMAL OBJECTIVE VALUE IN SUBPROBLEM J,
2397.      C      CORRESPONDING TO CURRENT ALLOCATION
2398.      C
2399.      IMPLICIT REAL*4 (A,C,E-H,O,P,K-W,Z), REAL*8 (B,D,X,Y),
2400.      1      INTEGER*4 (I-N,C)
2401.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
2402.      INTEGER*2 IPART,INCLMB,IVBND,IVID,ICBND
2403.      INTEGER*2 IZSTAR,IID,IBBND,IZBND
2404.      DOUBLE PRECISION E(1000)
2405.      REAL A(500)
2406.      COMMON/BLOCK2/ DFPART(122),REVBNB,INCLVAL,ICOL,LISTL,IVAL,IUIR,
2407.      1      NPVOT,IPTYPE,KCCST,IFEAS,IPART(122),INCUMB(122),
2408.      2      IVBND(500),IVID(500),IBBND(500)
2409.      COMMON/BLOCK3/ RLAMDA(5,20,10),ZNAUT(20,10),ZJBAR(10),NLINK,NSUBS,
2410.      1      IZINC,IZBAR,IUDM,IBLL,NJFIX,JFIXSM,IB(5),JFIX(10),NSOL(10),
2411.      2      NLAMDA(10),IBSTAR(5,40,10),IBUSED(5,40,10),IBUB(5,10),IBLB(5,10),
2412.      3      IZSTAR(40,10),IBID(2,200),IBBND(200),IZBND(200)
2413.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
2414.      1      OY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SJMNF,MSTAT,
2415.      2      IUBJ,IRCP,IVIN,IUOT,ITCNT,INVERQ,ITFRQ,ITSINV,IFFEZ,
2416.      3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
2417.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
2418.      INCUMB(1) = -10000
2419.      KSTORE = 0
2420.      LL = NSOL(J)
2421.      IF (LL .EQ. 0) RETURN
2422.      NEW = LL + 1
2423.      C      SEARCH PREVIOUS ALLOCATIONS FOR BEST LOWER BOUND ON ZJ
2424.      DO 100 K=1,LL
2425.      IF (INCUMB(1) .GE. IZSTAR(K,J)) GO TO 100
2426.      DO 50 I=1,NLINK
2427.      IF (IBUSED(I,K,J) .GT. IBSTAR(I,NEW,J)) GO TO 100
2428.      50 CONTINUE
2429.      C      BIJ* >= KTH BIJ USED: UPDATE INCUMB(1)
2430.      INCUMB(1) = IZSTAR(K,J)
2431.      KSTORE = K
2432.      100 CONTINUE
2433.      C
2434.      C      KSTORE INDEXES THE PREVIOUSLY EXAMINED ALLOCATION GIVING THE
2435.      C      GREATEST LOWER BOUND ON ZJ
2436.      IF (KSTORE .EQ. 0) RETURN
2437.      DO 200 I=1,NLINK
2438.      200 INCUMB(I+1) = IBSTAR(I,NEW,J) - IBUSED(I,KSTORE,J)
2439.      LL = NLINK + 1

```



```

2318.      DC 400 I=1,NLINK
2319.      IF (IESTAR(I,LPI,J) .GT. IBSTAR(I,K,J)) GO TO 500
2320.      400 CONTINUE
2321.      C          TEST AGAINST LOWER BOUNDS
2322.      DC 450 I=1,NLINK
2323.      IF (IBSTAR(I,LPI,J) .LT. IBUSED(I,K,J)) GO TO 500
2324.      450 CONTINUE
2325.      C
2326.      C          ALLOCATION DOMINATED
2327.      IDCM = 1
2328.      RETURN
2329.      C
2330.      500 CONTINUE
2331.      1000 CONTINUE
2332.      C
2333.      C          ALLOCATION NOT DOMINATED
2334.      IDCM = 0
2335.      RETURN
2336.      END
2337.      SUBROUTINE LODATA(J)
2338.      C
2339.      C          LOADS DATA FOR SUBPROBLEM J INTO APPROPRIATE STORAGE LOCATIONS
2340.      C          FOR SOLUTION BY SUBROUTINES NORMAL (LP) AND BANDB (ILP)
2341.      C
2342.      IMPLICIT REAL*4 (A,C,E-H,C,P,R-n,Z), REAL*8 (B,D,X,Y),
2343.      1      INTEGER*4 (I-N,L)
2344.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
2345.      INTEGER*2 IZSTAR,IEID,IBBND,IZBND
2346.      INTEGER*2 IAS,LAS
2347.      DOUBLE PRECISION E(1000)
2348.      REAL A(500)
2349.      COMMON/BLOCK3/ RLAMDA(5,20,10),ZNAUT(20,10),ZJBAR(10),NLINK,NSUBS,
2350.      1 IZINC,IZBAR,IDCM,IBLL,NJFIX,JFIXSM,IB(5),JFIX(10),NSOL(10),
2351.      2 NLAMDA(10),IBSTAR(5,40,10),IBUSED(5,40,10),IBUB(5,10),IBLB(5,10),
2352.      3 IZSTAR(40,10),IBID(2,200),IBBND(200),IZBND(200)
2353.      COMMON/BLOCK4/ XLBS(20,10),XUBS(20,10),AMATRS(1000),RHS(10,10),
2354.      1 NRCLS(10),NCOLS(10),JFIRST(11),IAS(1000),LAS(20,10)
2355.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
2356.      1 DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
2357.      2 IUBJ,IRCLP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
2358.      3 JCULP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NJETA,
2359.      4 LE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
2360.      NCCL = NCOLS(J)
2361.      NROW = NROWS(J)
2362.      NELEM = JFIRST(J+1) - JFIRST(J)
2363.      LL = JFIRST(J) - 1
2364.      C          LOAD A,IA,LA,XLB,XUB,B,JH,KINBAS
2365.      DO 10 I=1,NLELEM
2366.      A(I) = AMATRS(LL+1)
2367.      10 IA(I) = IAS(LL+1)
2368.      DO 20 K=1,NCCL
2369.      LA(K) = LAS(K,J)
2370.      XLB(K) = XLBS(K,J)
2371.      20 XUB(K) = XUBS(K,J)
2372.      LA(NCCL+1) = NELEM + 1
2373.      B(1) = 0.
2374.      LL = NSOL(J) + 1
2375.      DC 30 I=1,NLINK
2376.      K = IESTAR(I,LL,J)
2377.      30 B(I+1) = FLCAT(K)
2378.      K = NLINK + 1

```

```

2257. C          ROUND BIG'S (IF NECESSARY) TO INTEGER VALUES
2258. C
2259.      LL = NCLW + 2
2260.      KK = NLINK + 1
2261.      NRCLND = 0
2262.      DO 600 I=1,NLINK
2263.      DSUM = DFPART(I+1) + 0.01
2264.      DO 510 J=LL,NCLL,KK
2265. 510 DSUM = DSUM + DFPART(J)
2266.      ISUM = I*INT(DSUM)
2267.      IF (ISUM .EQ. 0) GO TO 560
2268.      NRCLND = 1
2269.      DO 550 II=1,ISUM
2270.      ICCL = 1 + 1
2271.      CP = DFPART(ICCL)
2272.      DO 520 J=LL,NCLL,KK
2273.      IF (DFPART(J) .LE. CP) GO TO 520
2274.      CP = DFPART(J)
2275.      ICCL = J
2276. 520 CONTINUE
2277.      IPART(ICCL) = IPART(ICCL) + 1
2278.      DFPART(ICCL) = 0.
2279. 550 CONTINUE
2280. 560 LL = LL + 1
2281. 600 CONTINUE
2282. C
2283. C          INSTALL NEW INCUMBENT
2284.      DO 590 J=1,NCOL
2285. 590 INCUMB(J) = IPART(J)
2286.      RETURN
2287.      END
2288.      SUBROUTINE COMCHK(J,K)
2289. C
2290. C          COMPARES CURRENT ALLOCATION (FROM SUBROUTINE ALLOC) WITH
2291. C          PREVIOUSLY EXAMINED ALLOCATIONS IN EACH UNFIXED SUBPROBLEM
2292. C          TO DETERMINE IF OPT. SOL. CORRESPONDING TO CURRENT ALLOCATION
2293. C          IS ALREADY KNOWN FOR ANY SUBPROBLEM
2294. C*****DESCRIPTION OF PARAMETERS*****
2295. C          J = PARAMETER INDEXING FIRST SUBPROBLEM FOUND FOR WHICH OPTIMAL
2296. C          SOLUTION CORRESPONDING TO CURRENT ALLOCATION IS ALREADY
2297. C          KNOWN (OUTPUT)
2298. C          K = PARAMETER INDEXING THE PREVIOUSLY EXAMINED ALLOCATION IN
2299. C          SUBPROBLEM J WHICH LEADS TO SAME OPT. SOL. AS WOULD
2300. C          CURRENT ALLOCATION (OUTPUT)
2301. C*****
2302. C
2303.      IMPLICIT REAL*4 (A,C,E-H,I,P,R-W,Z), REAL*8 (B,D,X,Y),
2304.      1      INTEGER*4 (I-N,Q)
2305.      INTEGER*2 IZSTAR,IEID,IBBND,IZBND
2306.      COMMON/BLOCKS/ KLAMDA(5,20,10),ZNAUT(20,10),ZJBAR(10),NLINK,NSUBS,
2307.      IZINC,IZBAR,IBCM,IBLL,NJFIX,JFIXSM,IB(5),JFIX(10),NSOL(10),
2308.      ZNLAMDA(10),IBSTAR(5,40,10),IBUSED(5,40,10),IBUB(5,10),IBLB(5,10),
2309.      3 IZSTAR(40,10),IBID(2,200),IBBND(200),IZBND(200)
2310.      DO 1000 J=1,NSUBS
2311.      IF (JFIX(J) .GT. 0) GO TO 1000
2312.      IF (NSOL(J) .EQ. 0) GO TO 1000
2313.      L = NSOL(J)
2314.      LPI = L + 1
2315.      DO 500 K=1,L
2316. C          TEST ALLOCATION AGAINST UPPER BOUNDS OF
2317. C          BIG IN KTH SOL. OF SUB J

```



```

2196.      ITEMF = ITEMF + N
2197.      500 CONTINUE
2198.      LA(NCCL+1) = NELEM + 1
2199.      ITSINV = 99999
2200.      INVRFC = NRCLW
2201.      RETURN
2202.      END
2203.      SUBROUTINE TESTZ
2204.      C
2205.      C      TESTS LP-OPTIMAL SOLUTION TO MASTER PROBLEM FOR FATHOMING AT
2206.      C      CURRENT NODE IN ALLOCATION-VARIABLES SEARCH TREE. IF NOT
2207.      C      FATHOMED, RUNS LP-OPTIMAL ALLOCATION TO OBTAIN AN INTEGER-
2208.      C      FEASIBLE ALLOCATION.
2209.      C
2210.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
2211.      1      INTEGER*4 (I-N,Q)
2212.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
2213.      INTEGER*2 IPART,INCLMB,IVBND,IVID,ICBND
2214.      INTEGER*2 IZSTAR,IEID,IBBND,IZBND
2215.      DOUBLE PRECISION E(1000)
2216.      REAL A(500)
2217.      COMMON/BLUCK/ ZTCLZE,ZTOLPV,ZTCCST,ZTOLRJ,ZTCLSM,QRO,QMA,QBA,QFI,
2218.      1      QEO,QBL,QA,QPL,QM1,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
2219.      2      QE,QF,QL,QC,QR,QM,GG,NTMAX
2220.      COMMON/BLUCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
2221.      1      NPVCT,IPTYPE,RCCST,IFEAS,IPART(122),INCUMB(122),
2222.      2      IVBND(500),IVID(500),IBBND(500)
2223.      COMMON/BLUCK3/ KLAMDA(5,20,10),ZNAUT(20,10),ZJBAR(10),NLINK,NSUBS,
2224.      1      IZINC,IZBAR,IDUM,IBLL,NJFIX,JFIXSM,IB(5),JFIX(10),NSOL(10),
2225.      2      NLAMDA(10),IBSTAR(5,40,10),IBUSED(5,40,10),IEUB(5,10),IBLB(5,10),
2226.      3      IZSTAR(40,10),IBID(2,200),IBBND(200),IZBND(200)
2227.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
2228.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
2229.      2      IOBJ,IRCLW,IVIN,IVOUT,ITCNT,INVRFC,ITRFRQ,ITSINV,IFFEZ,
2230.      3      JCULP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
2231.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
2232.      DP = X(IOBJ) + ZTOLZE
2233.      INCVAL = IDINT(DP)
2234.      IF (DP .LT. 0.) INCVAL = INCVAL - 1
2235.      IF (MSTAT .EQ. QN) INCVAL = -10000
2236.      C      TEST FOR FATHOMING
2237.      IF (INCVAL .LE. IZINC) RETURN
2238.      C
2239.      C      COMPUTE INTEGER AND FRACTIONAL PARTS OF EACH ZJ, BIJ
2240.      C
2241.      DO 100 J=1,NCOL
2242.      IF (KINBAS(J)) 40,50,60
2243.      40 DFPART(J) = 0.
2244.      IPART(J) = ICINT(XLB(J))
2245.      GO TO 100
2246.      50 DFPART(J) = 0.
2247.      IPART(J) = ICINT(XLB(J))
2248.      GO TO 100
2249.      60 I = KINBAS(J)
2250.      DP = X(I) + ZTOLZE
2251.      IPART(J) = IDINT(DP)
2252.      IF (DP .LT. 0.) IPART(J) = IPART(J) - 1
2253.      K = IPART(J)
2254.      DFPART(J) = X(I) - FLOAT(K)
2255.      100 CONTINUE
2256.      C

```



```

2135.      31ZSTAR(40,10),1B1D(2,200),1B5ND(200),1ZBND(200)
2136.      COMMON C,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
2137.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
2138.      2      IUBJ,IRCP,IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,ITSINV,IFFEZ,
2139.      3      JCULP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NJETA,
2140.      4      IE(1000),IA(500),LE(252),LAI(122),KINBAS(122),JH(60)
2141.      NROW = NLINK + 1
2142.      B(1) = 0.
2143.      C      INSTALL RHS FOR LINKING CONSTRAINTS
2144.      DO 10 I=2,NROW
2145.      ITEMP = 13(1-1)
2146.      10 B(I) = FLOAT(ITEMP)
2147.      C      COMPUTE NROW,NCOL
2148.      DO 100 J=1,NSUBS
2149.      NROW = NROW + NLAMDA(J)
2150.      100 CONTINUE
2151.      NCOL = NROW + NSUBS*(NLINK+1)
2152.      C
2153.      C      INSTALL SLACKS
2154.      C
2155.      DO 200 I=1,NROW
2156.      A(I) = 1.
2157.      LA(I) = 1
2158.      IA(I) = 1
2159.      KINBAS(I) = 1
2160.      JH(I) = 1
2161.      XLB(I) = 0.
2162.      200 XUB(I) = 10000.
2163.      C
2164.      C      INSTALL ZJ,ZIJ COLUMNS FOR UNFIXED SUBPROBLEMS
2165.      NELEM = NROW
2166.      ICCL = NROW
2167.      ITEMP = NLINK + 1
2168.      DO 500 J=1,NSUBS
2169.      NELEM = NELEM + 1
2170.      ICCL = ICCL + 1
2171.      A(NELEM) = -1.
2172.      IA(NELEM) = 1
2173.      LA(ICCL) = NELEM
2174.      N = NLAMDA(J)
2175.      DO 310 K=1,N
2176.      NELEM = NELEM + 1
2177.      A(NELEM) = 1.
2178.      II = ITEMP + K
2179.      IA(NELEM) = II
2180.      310 B(II) = DBLE(ZNAUT(K,J))
2181.      II = ITEMP + 1
2182.      KINBAS(ICCL) = II
2183.      KINBAS(II) = 0
2184.      JH(II) = ICCL
2185.      DO 350 I=1,NLINK
2186.      NELEM = NELEM + 1
2187.      ICCL = ICCL + 1
2188.      LA(ICCL) = NELEM
2189.      A(NELEM) = 1.
2190.      IA(NELEM) = 1 + I
2191.      DO 330 K=1,N
2192.      NELEM = NELEM + 1
2193.      A(NELEM) = -RLAMDA(I,K,J)
2194.      330 IA(NELEM) = ITEMP + K
2195.      350 KINBAS(ICCL) = 0

```

```

2074.      GO TO 8700
2075.      8700 DE = XLS(J)
2076.      8750 LL = LA(J)
2077.      KK = LA(J+1) - 1
2078.      DO 8800 I=LL, KK
2079.      IR = IA(I)
2080.      8800 Y(IR) = Y(IR) - A(I)*DE
2081.      9000 CONTINUE
2082.      CALL FTRAN(1)
2083.      CALL SHIFTR(3,2)
2084.      C      SOLVE MASTER PROBLEM AS LP
2085.      90 CALL NORMAL
2086.      C      TEST FOR FATHOMING; ROUND BIJ'S TO GET INTEGER-FEASIBLE SOL.
2087.      CALL TEST2
2088.      C
2089.      2000 IF (INCVL .LE. IZINC) RETURN
2090.      ICCL = NKOW + 1
2091.      C      COMPUTE LP-BOUND ON OBJ. VAL. IN UNFIXED SUBPROBLEMS FOR USE
2092.      C      IN SUBPROBLEM SELECTION RULE WHEN IPARM = 1
2093.      DO 18 J=1, NSUBS
2094.      IF (JFIX(J) .GT. 0) GO TO 18
2095.      I = KINGAS(ICCL)
2096.      ZJBAR(J) = SNGL(X(I))
2097.      18 ICCL = ICCL + NLINK + 1
2098.      C      INSTALL NEW ALLOCATION IN IBSTAR
2099.      IDIR = NKOW + 1
2100.      DO 2100 J=1, NSUBS
2101.      IF (JFIX(J) .GT. 0) GO TO 2100
2102.      LISTL = NSOL(J) + 1
2103.      DO 2050 I=1, NLINK
2104.      2050 IBSTAR(I, LISTL, J) = INCUMB(IDIR+1)
2105.      2100 IDIR = IDIR + NLINK + 1
2106.      DO 2200 I=1, NLINK
2107.      IPI = I + 1
2108.      IF (INCUMB(IPI) .EQ. 0) GO TO 2200
2109.      DO 2150 J=1, NSUBS
2110.      IF (JFIX(J) .GT. 0) GO TO 2150
2111.      LISTL = NSOL(J) + 1
2112.      IBSTAR(I, LISTL, J) = IBSTAR(I, LISTL, J) + INCUMB(IPI)
2113.      II = IBSTAR(I, LISTL, J) - IBUB(I, J)
2114.      IF (II .LE. 0) GO TO 2200
2115.      IBSTAR(I, LISTL, J) = IBUB(I, J)
2116.      INCUMB(IPI) = II
2117.      2150 CONTINUE
2118.      2200 CONTINUE
2119.      RETURN
2120.      END
2121.      SUBROUTINE LUMSTR
2122.      C
2123.      C      LOADS DATA FOR MASTER PROBLEM INTO APPROPRIATE STORAGE
2124.      C      LOCATIONS FOR SOLUTION BY SUBROUTINE NORMAL
2125.      C
2126.      IMPLICIT REAL*4 (A,C,E-H,C,P,R-W,Z), REAL*8 (B,D,X,Y),
2127.      1      INTEGER*4 (I-N,Q)
2128.      INTEGER*2 JH, KINGAS, LA, LE, IA, IE
2129.      INTEGER*2 IZSTAR, IBID, IBND, IZBND
2130.      DOUBLE PRECISION E(1000)
2131.      REAL A(500)
2132.      COMMON/BLCKB/ KLAUDA(5,20,10), ZNAUT(20,10), ZJBAR(10), NLINK, NSUBS,
2133.      1 IZINC, IZBAR, IZCM, IBL, NJFIX, JFIXSM, IB(5), JFIX(10), NSOL(10),
2134.      2 NLAMEA(10), IBSTAR(5,40,10), IBUSED(5,40,10), IBUB(5,10), IBLB(5,10),

```



```

2013. C REOPTIMIZED STARTING FROM PREVIOUSLY OPTIMAL SOLUTION
2014. C (I.E., DON'T CALL LDMSTR)
2015. C (*****
2016. C
2017. IMPLICIT REAL*4 (A,C,E-H,G,P,K-W,Z), REAL*8 (B,D,X,Y),
2018. 1 INTEGER*4 (I-N,Q)
2019. INTEGER*2 JH,KINBAS,LA,LE,IA,IE
2020. INTEGER*2 IPAKT,INCLMB,IVBND,IVID,ICBND
2021. INTEGER*2 IZSTAR,IEID,IBBND,IZBND
2022. DOUBLE PRECISION E(1000)
2023. REAL A(500)
2024. COMMON/BLCK/ ZTOLZE,ZTOLPV,ZTOLST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
2025. 1 GED,CBE,QA,QPL,QMI,CZ,QI,QF,CN,QU,NRMAX,NEMAX,QB,QC,
2026. 2 GE,QH,QL,QU,QR,QM,QG,ATMAX
2027. COMMON/BLCK2/ DFPART(122),REV8ND,INCVAL,ICCL,LISTL,IVAL,IDIR,
2028. 1 NPIVCT,IPTYPE,KCCST,IFEAS,IPART(122),INCUMB(122),
2029. 2 IVBND(500),IVID(500),ICBND(500)
2030. COMMON/BLCK3/ KLA,MDA(5,20,10),ZNAUT(20,10),ZJBAR(10),NLINK,NSUBS,
2031. IIZINC,IZBAR,IDCM,IBLL,NJFIX,JFIXSM,IB(5),JFIX(10),NSOL(10),
2032. 2NLAMDA(10),IBSTAR(5,40,10),IBUSED(5,40,10),IEUB(5,10),IBLB(5,10),
2033. 3IZSTAR(40,10),IEID(2,200),IBBND(200),IZBND(200)
2034. COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
2035. 1 UY,LE,UP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
2036. 2 IDBJ,IRGWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
2037. 3 JGULP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NJETA,
2038. 4 IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
2039. C IF NECESSARY, LOAD DATA FOR SOLUTION OF MASTER PROBLEM
2040. IF (IPAK.EQ. 0) CALL LDMSTR
2041. ICCL = NROW
2042. DO 50 J=1,NSUBS
2043. IF (JFIX(J).GT. 0) GO TO 40
2044. ICCL = ICCL + 1
2045. C INSTALL BOUNDS ON ZJ COLUMNS
2046. XLB(ICCL) = -10000.
2047. XUB(ICCL) = 10000.
2048. C INSTALL BOUNDS ON BIJ COLUMNS
2049. DO 30 I=1,NLINK
2050. ICCL = ICCL + 1
2051. II = IBLB(I,J)
2052. XLB(ICCL) = FLOAT(II)
2053. II = IEUB(I,J)
2054. XUB(ICCL) = FLOAT(II)
2055. 30 GO TO 50
2056. C INSTALL BOUNDS ON COLUMNS ASSOCIATED WITH FIXED SUBPROBLEMS
2057. 40 K = JFIX(J)
2058. ICCL = ICCL + 1
2059. II = IZSTAR(K,J)
2060. XLB(ICCL) = FLOAT(II)
2061. XUB(ICCL) = XLB(ICCL)
2062. DO 45 I=1,NLINK
2063. ICCL = ICCL + 1
2064. II = IBLB(I,J)
2065. XLB(ICCL) = FLOAT(II)
2066. 45 XUB(ICCL) = XLB(ICCL)
2067. 50 CCNTINC
2068. ITCNT = 0
2069. IF (IPAK.EQ. 0) GO TO 90
2070. CALL SHIFTR(1,3)
2071. DO 9000 J=1,NCOL
2072. IF (KINBAS(J))8600,8700,9000
2073. 8600 DE = XLB(J)

```



```

1952.      XLB(ICCL) = FLOAT(NTMP(2))
1953.      IF (KINBAS(ICCL) .LT. 0) GO TO 1000
1954.      KINBAS(ICCL) = -1
1955.      NTEMP(3) = 1
1956.      C
1957.      1000 IVIC(LISTL) = -IVIC(LISTL)
1958.      IVBND(LISTL) = NTEMP(1)
1959.      ICBND(LISTL) = -10000
1960.      C
1961.      C      UPDATE X
1962.      IF (NTEMP(3) .EQ. 0) RETURN
1963.      CALL SHIFTR(1,3)
1964.      DO 9000 J=1,NCLL
1965.      IF (KINBAS(J)) 8000,8700,9000
1966.      8000 DE = XLB(J)
1967.      GO TO 8750
1968.      8700 DE = XLB(J)
1969.      8750 LL = LA(J)
1970.      KK = LA(J+1) - 1
1971.      DO 8800 I=LL, KK
1972.      IR = IA(I)
1973.      8800 Y(IR) = Y(IR) - A(I)*DE
1974.      9000 CONTINUE
1975.      CALL FTRAN(1)
1976.      CALL SHIFTR(3,2)
1977.      RETURN
1978.      C
1979.      C      NODE FATHOMED: UPDATE VAR. BOUNDS AND BACKTRACK AGAIN
1980.      2000 ICCL = IVIC(LISTL)
1981.      IF (ICCL .LT. 0) GO TO 2100
1982.      C
1983.      NTEMP(1) = IVBND(LISTL)
1984.      IF (KINBAS(ICCL)) 2010,2050,2050
1985.      2010 NTEMP(3) = 1
1986.      DP = XLB(ICCL) - XLB(ICOL)
1987.      DY = FLOAT(NTMP(1)) - XLB(ICOL)
1988.      IF (DP .LT. DY) KINBAS(ICOL) = 0
1989.      2050 XLB(ICCL) = FLOAT(NTMP(1))
1990.      GO TO 3000
1991.      C
1992.      2100 ICCL = -ICOL
1993.      NTEMP(1) = IVBND(LISTL)
1994.      IF (KINBAS(ICOL)) 2150,2110,2150
1995.      2110 NTEMP(3) = 1
1996.      DY = XLB(ICOL) - FLOAT(NTMP(1))
1997.      DP = XLB(ICOL) - XLB(ICOL)
1998.      IF (DP .LT. DY) KINBAS(ICOL) = -1
1999.      2150 XLB(ICCL) = FLOAT(NTMP(1))
2000.      C
2001.      3000 LISTL = LISTL - 1
2002.      GO TO 50
2003.      C
2004.      END
2005.      SUBROUTINE ALLOCATE(IPAR)
2006.      C
2007.      C      SOLVES MASTER PROBLEM; REALLOCATES LINKING RESOURCE AMONG
2008.      C      UNFIXED SUBPROBLEMS
2009.      C*****DESCRIPTION OF PARAMETERS*****
2010.      C      IPAR = PARAMETER INDICATING WHETHER OR NOT ANY SUBPROBLEM HAS
2011.      C      BEEN EXPLICITLY SOLVED SINCE LAST MASTER PROBLEM
2012.      C      SOLUTION. IF NOT (IPAR=1), MASTER PROBLEM CAN BE

```

```

1891.      C
1892.      C          ICOL INDEXES BRANCHING VARIABLE CHOSEN
1893.      C          IDIR INDICATES BRANCHING DIRECTION CHOSEN
1894.      C
1895.      C          ADD OPPOSITE DIRECTION TO LIST
1896.      LISTL = LISTL + 1
1897.      IF (IDIR .EQ. -1) IVBND(LISTL) = IDINT(XLB(ICOL) + ZTOLZE)
1898.      IF (IDIR .EQ. 1) IVBND(LISTL) = IDINT(XUB(ICOL) + ZTOLZE)
1899.      IVID(LISTL) = IDIR*ICOL
1900.      ICBND(LISTL) = IVAL
1901.      C
1902.      C          REVISE BOUNDS ON BRANCHING VARIABLE FOR FORWARD DIRECTION
1903.      IF (IDIR .EQ. -1) XLB(ICOL) = DBLE(REVBND)
1904.      IF (IDIR .EQ. 1) XUB(ICOL) = DBLE(REVBND)
1905.      RETURN
1906.      END
1907.      SUBROUTINE BKTRAK
1908.      C
1909.      C          BACKTRACKS TO SELECT A PROMISING (UNFATHOMED) NODE FROM THE
1910.      C          LIST OF STORED NODES. LAST-IN-FIRST-OUT (LIFO) SELECTION RULE
1911.      C          IS EMPLOYED.
1912.      C
1913.      IMPLICIT REAL*4 (A,C,E-H,U,P,R-k,Z), REAL*8 (B,D,X,Y),
1914.      1      INTEGER*4 (I-N,Q)
1915.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1916.      INTEGER*2 IPART,INCLMB,IVBND,IVID,ICBND
1917.      DOUBLE PRECISION E(1000)
1918.      REAL A(500)
1919.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QF1,
1920.      1      QEO,CBL,QA,QPL,QM1,QZ,QI,QF,CN,QU,NRMAX,NEMAX,QB,QC,
1921.      2      QE,QF,QL,QO,QR,QM,QG,NTMAX
1922.      COMMON/BLOCK2/ DFPART(122),REVBND,INCLVAL,ICOL,LISTL,IVAL,IDIR,
1923.      1      NPVCT,IPTYPE,RCCST,IFEAS,IPART(122),INCUMB(122),
1924.      2      IVBND(500),IVID(500),IOBND(500)
1925.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPRUD,
1926.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1927.      2      IOBJ,IRCMP,IVIN,IVOUT,ITCNT,INVERQ,ITRFRQ,ITSINV,IFFEZ,
1928.      3      JCULP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1929.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1930.      NTEMP(3) = 0
1931.      C          IF LIST IS EMPTY, RETURN (COMPUTATIONS COMPLETED)
1932.      50 IF (LISTL .EQ. 0) RETURN
1933.      IF (ICBND(LISTL) .LE. INCLVAL) GO TO 2000
1934.      C
1935.      C          GET NEXT NODE FROM LIST
1936.      ICOL = IVID(LISTL)
1937.      IF (ICOL .LT. 0) GO TO 100
1938.      C
1939.      NTEMP(1) = IDINT(XLB(ICOL) + ZTOLZE)
1940.      NTEMP(2) = IVBND(LISTL)
1941.      XLB(ICOL) = XUB(ICOL) + 1.
1942.      XUB(ICOL) = FLGAT(NTEMP(2))
1943.      IF (KINBAS(ICOL) .GT. 0) GO TO 1000
1944.      KINBAS(ICOL) = 0
1945.      NTEMP(3) = 1
1946.      GO TO 1000
1947.      C
1948.      100 ICOL = -ICOL
1949.      NTEMP(1) = IDINT(XUB(ICOL) + ZTOLZE)
1950.      NTEMP(2) = IVBND(LISTL)
1951.      XUB(ICOL) = XLB(ICOL) - 1.

```

```

1830.      C      DETERMINE BASIC VAR. X(IROWP) WITH MAX. UP- OR DOWN-PENALTY
1831.      DO 4900 I=1,NROW
1832.      IF (JF(I) .LE. NROW) GO TO 4900
1833.      IF (PL(I) .GT. PC(I)) GO TO 4600
1834.      IF (PC(I) .LE. PEN) GO TO 4900
1835.      PEN = PC(I)
1836.      IRCWP = I
1837.      IDIR = -1
1838.      NTEMP(1) = IPART(I) + 1
1839.      REVBND = FLCAT(NTEMP(1))
1840.      GO TO 4900
1841. 4600 IF (PL(I) .LE. PEN) GO TO 4900
1842.      PEN = PL(I)
1843.      IRCWP = I
1844.      IDIR = 1
1845.      NTEMP(1) = IPART(I)
1846.      REVBND = FLCAT(NTEMP(1))
1847. 4900 CONTINUE
1848.      IF (IRCWP .GT. 0) GO TO 4950
1849.      C      EACH UP- AND DOWN-PENALTY .EQ. 0. (DUAL-DEGENERACY). CHOOSE
1850.      C      ANY NONINTEGER BASIC VAR. AS BRANCHING VAR.
1851.      DO 4910 IRCWP=1,NROW
1852.      IF (JF(IRCWP) .LE. NROW) GO TO 4910
1853.      IF (OPPART(IRCWP) .GE. ZTOLZE) GO TO 4920
1854. 4910 CONTINUE
1855. 4920 PEN = PU(IRCWP)
1856.      IDIR = 1
1857.      NTEMP(1) = IPART(IRCWP)
1858.      REVBND = FLCAT(NTEMP(1))
1859. 4930 ICCL = JH(IRCWP)
1860.      DP = X(ICCL) - PEN + ZTOLZE
1861.      NTEMP(1) = ICENT(DP)
1862.      IF (CF .LT. 0.) NTEMP(1) = NTEMP(1) - 1
1863.      IF (IVAL .GT. NTEMP(1)) IVAL = NTEMP(1)
1864.      C      BRANCH ON CHOSEN VAR.
1865.      CALL BRANCH
1866. 5000 IF (IDIR .EQ. -1) IPTYPE = 0
1867.      IF (IDIR .EQ. 1) IPTYPE = -1
1868.      RETURN
1869.      END
1870.      SUBROUTINE BRANCH
1871.      C
1872.      C      BRANCHES ON VARIABLE X(ICOL) AS DETERMINED IN SUBROUTINE PENLTS
1873.      C
1874.      IMPLICIT REAL*4 (A,C,E-H,G,P,R-X,Z), REAL*8 (B,D,X,Y),
1875.      1      INTEGER*4 (I-N,Q)
1876.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1877.      INTEGER*2 IPART,INCLMB,IVBND,IVID,ICBND
1878.      DOUBLE PRECISION E(1000)
1879.      REAL Z(500)
1880.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTCLSM,CRO,QMA,QJA,QF1,
1881.      1      QEO,CBL,QA,CPL,QMI,CZ,QI,QF,CN,CU,NRMAX,NEMAX,QB,QC,
1882.      2      CE,QH,QL,QQ,QK,QM,CG,NTMAX
1883.      COMMON/BLOCK2/ DPART(122),REVBND,INCVL,ICOL,LISTL,IVAL,IDIR,
1884.      1      NPIVOT,IPTYPE,KCCST,IFEAS,IPART(122),INCUMB(122),
1885.      2      IVBND(500),IVID(500),ICBND(500)
1886.      COMMON E,XLE(122),XCB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
1887.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1888.      2      IOBJ,IRCWP,IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,ITSINV,IFFEZ,
1889.      3      JCULP,NROW,NCOL,NELEM,NETA,NLELEN,NLETA,NUELEM,NUETA,
1890.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)

```



```

1769.      200 DP = [ABS(Y(I))
1770.      NTEMP(1) = IDINT(DP)
1771.      DP = DP - FLOAT(NTEMP(1))
1772.      IF ((DP .LE. ZTOLZE) .OR. (DP .GE. 1.-ZTOLZE)) GO TO 500
1773.      IF (Y(I) .LT. 0.) DP = 1. - DP
1774.      IF (DP .GT. DEPART(I)) GO TO 310
1775.      DE = Y(IUBJ)*DEPART(I)/DP
1776.      GO TO 320
1777.      310 DE = Y(IUBJ)*(1. - DEPART(I))/(1. - DP)
1778.      320 IF (DE .LT. PG(I)) PG(I) = DE
1779.      C
1780.      500 CONTINUE
1781.      1000 CONTINUE
1782.      C
1783.      C          COMPUTE LARGEST GUMORY PENALTY AND TEST FOR FATHOMING
1784.      PEN = 0.
1785.      DO 2000 I=1,NROW
1786.      IF (JF(I) .LE. NROW) GO TO 2000
1787.      IF (PG(I) .GT. PEN) PEN = PG(I)
1788.      2000 CONTINUE
1789.      DP = X(IUBJ) - PEN + ZTOLZE
1790.      IVAL = IDINT(DP)
1791.      IF (DP .LT. 0.) IVAL = IVAL - 1
1792.      IF (IVAL .GT. INCVAL) GO TO 3000
1793.      IDIR = 0
1794.      RETURN
1795.      C
1796.      C          PROBLEM NOT FATHOMED: CHECK FOR FORCED BRANCHES ON X(I)
1797.      3000 NTEMP(2) = 0
1798.      DO 3500 I=1,NROW
1799.      IF (JF(I) .LE. NROW) GO TO 3500
1800.      IF (PU(I) .GT. PD(I)) GO TO 3600
1801.      DP = X(IUBJ) - PD(I) + ZTOLZE
1802.      NTEMP(1) = IDINT(DP)
1803.      IF (DP .LT. 0.) NTEMP(1) = NTEMP(1) - 1
1804.      IF (NTEMP(1) .GT. INCVAL) GO TO 3500
1805.      C          FORCED BRANCH UP ON X(I)
1806.      IVAL = NTEMP(1)
1807.      IDIR = -1
1808.      NTEMP(1) = IPART(I) + 1
1809.      GO TO 3700
1810.      C
1811.      3600 DP = X(IUBJ) - PU(I) + ZTOLZE
1812.      NTEMP(1) = IDINT(DP)
1813.      IF (DP .LT. 0.) NTEMP(1) = NTEMP(1) - 1
1814.      IF (NTEMP(1) .GT. INCVAL) GO TO 3500
1815.      C          FORCED BRANCH DOWN ON X(I)
1816.      IVAL = NTEMP(1)
1817.      IDIR = 1
1818.      NTEMP(1) = IPART(I)
1819.      3700 IRCWP = I
1820.      ICCL = JH(IRCWP)
1821.      REVEN = FLOAT(NTEMP(1))
1822.      NTEMP(2) = 1
1823.      CALL BRANCH
1824.      3900 CONTINUE
1825.      IF (NTEMP(2) .GT. 0) GO TO 5000
1826.      C
1827.      C          NO FORCED BRANCHES: CHOOSE BRANCHING VAR. AND DIRECTION
1828.      PEN = 0.
1829.      IRCWP = 0

```

```

1708.      1      LEO,CBO,QA,QPL,QMI,QZ,WI,QF,CN,CU,NRMAX,NEMAX,QB,QC,
1709.      2      GE,QF,QL,QU,QK,QM,CG,NTMAX
1710.      COMMON/BLCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1711.      1      NPivot,IPTYPE,RCOST,IFEAS,IPART(122),INCJMB(122),
1712.      2      IVBND(500),IVID(500),ICBND(500)
1713.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
1714.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1715.      2      IOBJ,IREXP,IVIN,IVOUT,ITCNT,INVERQ,ITRFRQ,ITSINV,IFFEZ,
1716.      3      JCCLP,NROW,NCOL,NLEEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1717.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1718.      DO 10 I=1,NROW
1719.      IF (DFPART(I) .LT. ZTOLZE) GO TO 5
1720.      PU(I) = 1.E6
1721.      PD(I) = 1.E6
1722.      PG(I) = 1.E6
1723.      GO TO 10
1724.      5 PU(I) = 0.
1725.      PD(I) = 0.
1726.      PG(I) = 0.
1727.      10 CONTINUE
1728.      C
1729.      DO 1000 J=1,NCOL
1730.      IF (KINBAS(J) .GT. 0) GO TO 1000
1731.      IF ((XUB(J) - XLB(J)) .LE. ZTOLZE) GO TO 1000
1732.      CALL UNPACK(J)
1733.      CALL FTRAN(I)
1734.      IF (KINBAS(J) .EQ. 0) GO TO 30
1735.      DO 20 I=1,NROW
1736.      20 Y(I) = -Y(I)
1737.      C
1738.      C      CHECK FOR FORCED BRANCH ON XJ
1739.      30 IF (J .LE. NROW) GO TO 50
1740.      DP = X(IOBJ) - Y(ICBJ) + ZTOLZE
1741.      IVAL = IDINT(DP)
1742.      IF (DP .LT. 0.) IVAL = IVAL - 1
1743.      IF (IVAL .GT. INCVAL) GO TO 50
1744.      IDIR = 2*KINBAS(J) + 1
1745.      IF (IDIR .EQ. -1) REVBND = SNGL(XUB(J))
1746.      IF (IDIR .EQ. 1) REVBND = SNGL(XLB(J))
1747.      ICCL = J
1748.      CALL BRANCH
1749.      GO TO 1000
1750.      C
1751.      50 DO 500 I=1,NROW
1752.      IF (JH(I) .LE. NROW) GO TO 500
1753.      IF (DFPART(I) .LT. ZTOLZE) GO TO 500
1754.      C
1755.      C      COMPUTE UP PENALTY FOR X(I), XJ
1756.      100 IF (Y(I) .GT. -ZTOLPV) GO TO 200
1757.      DE = Y(IOBJ)*(DFPART(I) - 1.)/Y(I)
1758.      IF (DE .LT. Y(IOBJ)) DE = Y(IOBJ)
1759.      IF (DE .LT. PU(I)) PU(I) = DE
1760.      GO TO 300
1761.      C
1762.      C      COMPUTE DOWN PENALTY FOR X(I), XJ
1763.      200 IF (Y(I) .LT. ZTOLPV) GO TO 300
1764.      DE = Y(IOBJ)*DFPART(I)/Y(I)
1765.      IF (DE .LT. Y(IOBJ)) DE = Y(IOBJ)
1766.      IF (DE .LT. PD(I)) PD(I) = DE
1767.      C
1768.      C      COMPUTE GUMCKY PENALTY FOR X(I), XJ

```

```

1647.          IVAL = IDINT(DP)
1648.          IF (DP .LT. 0.) IVAL = IVAL - 1
1649.          IF ((MSTAT .EQ. 0N) .OR. (IVAL .LE. INCVAL)) GO TO 2000
1650.      C
1651.      C          COMPUTE INTEGER AND FRACTIONAL PARTS OF EACH BASIC VAR.
1652.      C
1653.      DO 100 I=1,NROW
1654.          IPART(I) = IDINT(X(I) + ZTOLZE)
1655.          NTEMP(I) = IPART(I)
1656.      100 DFPART(I) = X(I) - FLOAT(NTEMP(I))
1657.      C
1658.      C          CHECK FOR ALL-INTEGER SOLUTION
1659.      C
1660.      DO 200 I=1,NROW
1661.          IF (JF(I) .LE. NROW) GO TO 200
1662.          IF (DFPART(I) .GE. ZTOLZE) RETURN
1663.      200 CONTINUE
1664.      C
1665.      C          SOLUTION ALL-INTEGER: INSTALL AS NEW INCUMBENT
1666.      C
1667.          IFEAS = 1
1668.          INCVAL = IVAL
1669.          DO 1000 J=1,NCOL
1670.              IF (KINBAS(J))600,700,1000
1671.          600 INCLMB(J) = IDINT(XUB(J))
1672.              GO TO 1000
1673.          700 INCLMB(J) = IDINT(XLB(J))
1674.          1000 CONTINUE
1675.      C
1676.          DO 1100 I=1,NROW
1677.              ICOL = JH(I)
1678.              INCLMB(ICOL) = IPART(I)
1679.          1100 CONTINUE
1680.      C
1681.      C          CURRENT PROBLEM NO LONGER OF INTEREST
1682.      C
1683.      2000 MSTAT = Q1
1684.          RETURN
1685.      END
1686.      SUBROUTINE PENLTS
1687.      C
1688.      C          COMPUTES TOMLIN'S IMPROVED UP- AND DOWN-PENALTIES AND THE
1689.      C          GCMCRY PENALTY FOR EACH NONINTEGER BASIC VARIABLE. ALSO CHECKS
1690.      C          FOR FORCED BRANCHES ON BOTH BASIC AND NONBASIC VARIABLES. IN
1691.      C          THE ABSENCE OF FORCED BRANCHES ON BASIC VARIABLES, THE
1692.      C          BRANCHING VARIABLE IS CHOSEN TO BE THE ONE WITH LARGEST
1693.      C          ASSOCIATED UP- OR DOWN-PENALTY. THE FORWARD BRANCH IS MADE IN
1694.      C          THE DIRECTION OPPOSITE TO THE MAXIMUM PENALTY. THE NODE
1695.      C          CORRESPONDING TO THE BRANCH IN THE SAME DIRECTION AS THE
1696.      C          MAXIMUM PENALTY IS ADDED TO THE LIST, TO BE EXAMINED LATER.
1697.      C          THE BRANCHING PROCESS ITSELF IS CARRIED OUT IN SUBROUTINE
1698.      C          BRANCH, WHICH IS CALLED FROM SUBROUTINE PENLTS.
1699.      C
1700.      IMPLICIT REAL*4 (A,C,E-H,U,P,R-W,Z), REAL*8 (B,D,X,Y),
1701.      1      INTEGER*4 (I-N,Q)
1702.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1703.      INTEGER*2 IPART,INCLMB,IVBND,IVID,ICBND
1704.      DOUBLE PRECISION E(1000)
1705.      REAL A(500)
1706.      REAL PU(60),PD(60),PG(60)
1707.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTOLST,ZTOLRJ,ZTOLSM,CRO,QMA,QBA,QFI,

```



```

1586.      CALL UNPACK(JCOLP)
1587.      CALL FIRAN(1)
1588.      ICCL = JH(IROWP)
1589.      DP = (X(IROWP) - XLB(ICOL))/Y(IROWP)
1590.      GO TO 2000
1591.  C
1592.  C          LEAVING VARIABLE EXCEEDS ITS UPPER BOUND
1593.  C
1594.      1000 DP = 1.E10
1595.      DO 1500 J=1,NCCL
1596.          IF (KINBAS(J) .GT. 0) GO TO 1500
1597.          IF ((XUB(J) - XLB(J)) .LE. ZTOLZE) GO TO 1500
1598.          CALL UNPACK(J)
1599.          CALL FIRAN(1)
1600.          IF (KINBAS(J) .EQ. -1) GO TO 1200
1601.          IF (Y(IROWP) - ZTOLPV) 1500,1225,1225
1602.      1200 IF (Y(IROWP) + ZTOLPV) 1225,1225,1500
1603.  C
1604.      1225 DE = Y(ICBJ)/Y(IROWP)
1605.          IF (DE - DP) 1250,1500,1500
1606.      1250 JCOLP = J
1607.          DP = DE
1608.      1500 CONTINUE
1609.  C
1610.      IF (JCOLP .EQ. 0) RETURN
1611.      CALL UNPACK(JCOLP)
1612.      CALL FIRAN(1)
1613.      ICCL = JH(IROWP)
1614.      DP = (X(IROWP) - XUB(ICOL))/Y(IROWP)
1615.  C
1616.      2000 IF (KINBAS(JCOLP) .EQ. 0) DE = DP + XLB(JCOLP)
1617.          IF (KINBAS(JCOLP) .EQ. -1) DE = DP + XUB(JCOLP)
1618.      NPIVCT = 1
1619.      RETURN
1620.      END
1621.      SUBROUTINE TESTX
1622.  C          TESTS LP-OPTIMAL SOLUTION AT CURRENT NODE FOR FATHOMING.
1623.  C          FATHOMING OCCURS IF
1624.  C          (1) LP PROBLEM AT CURRENT NODE IS INFEASIBLE (MSTAT = QN); OR
1625.  C          (2) LP-OPTIMAL OBJECTIVE VALUE (IVAL) .LE. OBJECTIVE VALUE OF
1626.  C          CURRENT INCUMBENT SOLUTION (INCVAL); OR
1627.  C          (3) LP-OPTIMAL SOLUTION SATISFIES INTEGER RESTRICTIONS.
1628.  C
1629.      IMPLICIT REAL*4 (A,C,E-H,I,P,R-X,Z), REAL*8 (B,D,X,Y),
1630.      1      INTEGER*4 (I-N,Q)
1631.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1632.      INTEGER*2 IPART,INCUMB,IVBND,IVIC,ICBND
1633.      DOUBLE PRECISION E(1000)
1634.      REAL A(500)
1635.      COMMON/BLCK1/ ZTOLZE,ZTOLPV,ZTOLST,ZTOLRJ,ZTOLSM,QKO,QMA,QBA,QFI,
1636.      1      QEO,QBL,QA,QPL,QM1,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1637.      2      QE,QF,QL,QO,QR,QM,QG,NTMAX
1638.      COMMON/BLCK2/ DFPART(122),REV8ND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1639.      1      NPIVCT,IPTYPE,RCCST,IFEAS,IPART(122),INCUMB(122),
1640.      2      IVBND(500),IVID(500),ICBND(500)
1641.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),OSUM,DPROD,
1642.      1      OY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1643.      2      IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,ITSINV,IFFE2,
1644.      3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1645.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1646.      DP = X(IOBJ) + ZTOLZE

```

```

1525. C      BASIC VARIABLE CN ROW I FALLS BELOW ITS LOWER BOUND
1526. 1CC DE = XLB(ICCL) - X(I)
1527. IF (DE .LE. LP) GO TO 1000
1528. IPTYPE = 0
1529. GO TO 250
1530. C
1531. C      BASIC VARIABLE CN ROW I EXCEEDS ITS UPPER BOUND
1532. 2CC DE = X(I) - XUB(ICCL)
1533. IF (DE .LE. DP) GO TO 1000
1534. IPTYPE = -1
1535. C
1536. 25C IROWP = I
1537. 1CCC CONTINUE
1538. RETURN
1539. END
1540. SUBROUTINE CCHLZC
1541. C
1542. C      SELECTS PIVOT COLUMN JCCLP FOR CURRENT DUAL-SIMPLEX ITERATION.
1543. C      SETS JCCLP = 0 IF LP-PROBLEM AT CURRENT NCDE IS INFEASIBLE.
1544. C      OTHERWISE, CHOOSES JCCLP TO MAINTAIN PRIMAL-OPTIMALITY
1545. C
1546. IMPLICIT REAL*4 (A,C,E-H,C,P,R-W,Z), REAL*8 (B,D,X,Y),
1547. 1 INTEGER*4 (I-N,Q)
1548. INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1549. INTEGER*2 IPART,INCLMB,IVBND,IVID,ICBND
1550. DOUBLE PRECISION E(1000)
1551. REAL A(500)
1552. COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZCCST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1553. 1 QEO,CBL,QA,QPL,QMI,CZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1554. 2 CE,QF,QL,QO,QR,QM,QG,NTMAX
1555. COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IUIR,
1556. 1 NPIVCT,IPTYPE,RCCST,IFEAS,IPART(122),INCUMB(122),
1557. 2 IVBND(500),IVID(500),IOBND(500)
1558. COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
1559. 1 DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1560. 2 IOBJ,IROWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
1561. 3 JCCLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1562. 4 IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1563. C
1564. JCCLP = 0
1565. IF (IPTYPE .EQ. -1) GO TO 1000
1566. C
1567. C      LEAVING VARIABLE FALLS BELOW ITS LOWER BOUND
1568. C
1569. CP = -1.E10
1570. DO 50C J=1,NCOL
1571. IF (KINBAS(J) .GT. 0) GO TO 500
1572. IF ((XUB(J) - XLB(J)) .LE. ZTOLZE) GO TO 500
1573. CALL CNPACK(J)
1574. CALL FIKAN(1)
1575. IF (KINBAS(J) .EQ. -1) GO TO 20C
1576. IF (Y(IROWP) + ZTOLPV) 225,225,500
1577. 2CC IF (Y(IROWP) - ZTOLPV) 500,225,225
1578. C
1579. 225 DE = Y(IOBJ)/Y(IROWP)
1580. IF (DE - DP) 500,500,250
1581. 25C JCCLP = J
1582. CP = CE
1583. 50C CONTINUE
1584. C
1585. IF (JCCLP .EQ. 0) RETURN

```



```

1464.      1000 CALL INVERT
1465.          ITSINV = 0
1466.      C
1467.          C          DUAL SIMPLEX CYCLE
1468.      C
1469.      C          CHOOSE PIVOT ROW IROWP
1470.      300 CALL ECHUZR
1471.          IF (IRCWP .GT. 0) GO TO 400
1472.          MSTAT = QBL
1473.          GO TO 100
1474.      C          CHOOSE PIVOT COLUMN JCCLP
1475.      400 CALL ECHUZR
1476.          IF (JCCLP .EQ. 0) GO TO 150
1477.          IVIN = JCCLP
1478.      C          UPDATE RIGHT-HAND SIDES TO REFLECT NEW BASIS RESULTING FROM
1479.      C          CURRENT SIMPLEX PIVOT
1480.      CALL LPBETA
1481.          ITCNT = ITCNT + 1
1482.          ITSINV = ITSINV + 1
1483.      C          DECIDE WHETHER TO REINVERT CURRENT BASIS
1484.          IF ((NELEM .GT. 5000) .OR. (ITSINV .GE. INVFRQ)) GO TO 1000
1485.      C          REINVERSION NOT NECESSARY YET; WRITE OUT NEW ETA-VECTOR FOR
1486.      C          CURRENT SIMPLEX PIVOT
1487.      CALL WETA
1488.          GO TO 300
1489.      END
1490.      SUBROUTINE ECHUZR
1491.      C
1492.      C          SELECTS PIVOT ROW IROWP FOR CURRENT DUAL-SIMPLEX ITERATION.
1493.      C          SETS IROWP = 0 IF CURRENT BASIS IS OPTIMAL. OTHERWISE, IROWP
1494.      C          IS CHOSEN TO BE THE ROW WITH GREATEST PRIMAL-INFEASIBILITY
1495.      C
1496.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1497.      1      INTEGER*4 (I-N,Q)
1498.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1499.      INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
1500.      DOUBLE PRECISION E(1000)
1501.      REAL A(500)
1502.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCST,ZTOLRJ,ZTOLSM,CRO,QMA,QBA,QFI,
1503.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1504.      2      QE,QH,QL,QU,QR,QM,QG,NTMAX
1505.      COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1506.      1      NPIVOT,IPTYPE,RCCST,IFEAS,IPART(122),INCUMB(122),
1507.      2      IVBND(500),IVID(500),ICBND(500)
1508.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
1509.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1510.      2      IOBJ,IRCWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
1511.      3      JCCLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1512.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1513.      C
1514.      C          CHOOSE ROW WITH GREATEST INFEASIBILITY
1515.      C
1516.      IRCWP = 0
1517.      DP = -1.E10
1518.      DO 1000 I=1,NROW
1519.          IF (I .EQ. IOBJ) GO TO 1000
1520.          ICCL = JH(I)
1521.          IF (X(I) .LT. (XLB(ICOL) - ZTOLZE)) GO TO 100
1522.          IF (X(I) .GT. (XUB(ICOL) + ZTOLZE)) GO TO 200
1523.          GO TO 1000
1524.      C

```



```

1403.      GO TO 6000
1404.      3000 IVIN = JCCLP
1405.      CALL UNPACK(JCCLP)
1406.      CALL FTRAN(1)
1407.      CALL CHUZR
1408.      CALL LPBETA
1409.      ITCNT = ITCNT + 1
1410.      ITSINV = ITSINV + 1
1411.      IF (NPIVCT .EQ. 0) GO TO 4010
1412.      IF (NELEM .GT. 5000) GO TO 1000
1413.      CALL WRETA
1414.      4010 IF (ITSINV .GE. INVFRQ) GO TO 1000
1415.      IF (ITCNT .GE. ITRFRQ) GO TO 6000
1416.      GO TO 1500
1417.      C
1418.      6000 RETURN
1419.      END
1420.      SUBROUTINE BANDB(INITBD)
1421.      C
1422.      C      MASTER PROGRAM FOR BRANCH-AND-BOUND INTEGER PROGRAMMING ROUTINE
1423.      C      ALSO SERVES AS MASTER PROGRAM FOR REOPTIMIZATION VIA DUAL-
1424.      C      SIMPLEX METHOD AFTER A FORWARD BRANCH,
1425.      C*****DESCRIPTION OF PARAMETERS*****
1426.      C      INITBD = INITIAL LOWER BOUND ON MAXIMAL OBJECTIVE VALUE (INPUT)
1427.      C*****
1428.      C
1429.      IMPLICIT REAL*4 (A,C,E-H,G,P,R-h,Z), REAL*8 (B,D,X,Y),
1430.      1      INTEGER*4 (I-N,Q)
1431.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1432.      INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
1433.      DOUBLE PRECISION E(1000)
1434.      REAL A(500)
1435.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCCST,ZTOLRJ,ZTCLSM,QRO,QMA,QBA,QFI,
1436.      1      QEO,CBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1437.      2      GE,QF,CL,QQ,QR,QM,QG,NTMAX
1438.      COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1439.      1      NPIVCT,IPTYPE,RCCST,IFEAS,IPART(122),INCUMB(122),
1440.      2      IVBND(500),IVID(500),IGBND(500)
1441.      COMMON E,XLB(122),XLB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
1442.      1      DY,DE,UP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1443.      2      IOBJ,IRCWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
1444.      3      JCCLP,NROW,NCOL,NELEM,ETA,NLELEM,NLETA,NUELEM,NUETA,
1445.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1446.      LISTL = 0
1447.      INCVAL = INITBD
1448.      C      TEST FOR FATHOMING
1449.      100 CALL TESTA
1450.      IF (MSTAT .EQ. QBL) GO TO 200
1451.      C      CURRENT NODE FATHOMED; BACKTRACK TO LAST PROMISING NODE ON LIST
1452.      150 CALL BKTRK
1453.      C      IF LIST IS EMPTY, RETURN TO MAIN (COMPUTATIONS COMPLETED)
1454.      IF (LISTL .EQ. 0) RETURN
1455.      C      USE PRIMAL-SIMPLEX METHOD FOR REOPTIMIZATION AT NEW NODE
1456.      CALL NORMAL
1457.      GO TO 100
1458.      C      CURRENT NODE NOT FATHOMED; COMPUTE PENALTIES
1459.      C      BRANCHING AT CURRENT NODE IS DONE FROM SUBROUTINE PENLTS.
1460.      200 CALL PENLTS
1461.      IF (ICIR) 400,150,400
1462.      C
1463.      C      REINVERT CURRENT BASIS

```

```

1342.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
1343.      1      DY,LE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1344.      2      IOBJ,IRCWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
1345.      3      JCULP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1346.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1347.      C
1348.      DC 1000 I=1,AROW
1349.      1000 X(I) = X(I) - Y(I)*DP
1350.      IF (NPIVOT .EQ. 1) GO TO 2000
1351.      KINBAS(JCULP) = -(KINBAS(JCULP) + 1)
1352.      IVCLT = JCULP
1353.      RETURN
1354.      2000 X(IRCWP) = DE
1355.      IVOUT = JH(IRCWP)
1356.      KINBAS(JCULP) = IRCWP
1357.      KINBAS(IVCLT) = IPTYPE
1358.      JH(IRCWP) = JCULP
1359.      RETURN
1360.      END
1361.      SUBROUTINE NORMAL
1362.      C
1363.      C      SERVES AS MASTER PROGRAM FOR LINEAR PROGRAMMING COMPONENT
1364.      C      (REVISED PRIMAL-SIMPLEX METHOD).
1365.      C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
1366.      C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1367.      C
1368.      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1369.      1      INTEGER*4 (I-N,Q)
1370.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1371.      INTEGER*2 IPART,INCLMB,IVBND,IVID,IOBND
1372.      DOUBLE PRECISION E(1000)
1373.      REAL A(500)
1374.      C
1375.      COMMON/BLOCK1/ ZTOLZE,ZTOLPV,ZTCCST,ZTOLRJ,ZTCLSM,QRO,QMA,QBA,QFI,
1376.      1      QEO,CBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1377.      2      QE,QH,QL,QU,QR,QM,CG,NTMAX
1378.      C
1379.      COMMON/BLOCK2/ DEPART(122),REVBND,INCLVAL,ICOL,LISTL,IVAL,IDIR,
1380.      1      NPIVCT,IPTYPE,RCCST,IFEAS,IPART(122),INCLMB(122),
1381.      2      IVBND(500),IVID(500),IOBND(500)
1382.      C
1383.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
1384.      1      DY,LE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1385.      2      IOBJ,IRCWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
1386.      3      JCULP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1387.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1388.      C
1389.      IF (ITSINV .LT. INVFRQ) GO TO 1500
1390.      1000 CALL INVERT
1391.      ITSINV = 0
1392.      C
1393.      C      SIMPLEX CYCLE
1394.      C
1395.      1500 CALL FORMC
1396.      CALL BTRAN
1397.      CALL PRICE
1398.      IF (JCULP .GT. 0) GO TO 3000
1399.      IF (MSTAT .EQ. Q1) GO TO 2000
1400.      MSTAT = QBL
1401.      GO TO 6000
1402.      2000 MSTAT = QN

```



```

1281. C SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
1282. C BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1283. C
1284. C IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1285. 1 INTEGER*4 (I-N,Q)
1286. C INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1287. C DOUBLE PRECISION E(1000)
1288. C REAL A(500)
1289. C
1290. C COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1291. 1 QEO,QBL,QA,QPL,QMI,QZ,QI,QF,CN,QU,NRMAX,NEMAX,QB,QC,
1292. 2 QE,QH,QL,QO,QR,QM,QG,NTMAX
1293. C
1294. C COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,OPRUD,
1295. 1 DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1296. 2 IOBJ,IRCP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
1297. 3 JCULP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1298. 4 IE(1000),IA(500),LE(252),LAI(122),KINBAS(122),JH(60)
1299. C
1300. C SHIFT IE AND E OF J ELEMENTS
1301. C
1302. C NF = NEMAX - NLELEM + 1
1303. C INCR = 0
1304. C DO 1000 I = NF,NEMAX
1305. C INCR = INCR + 1
1306. C IE(NLELEM + INCR) = IE(I)
1307. C E(NLELEM + INCR) = E(I)
1308. C 1000 CONTINUE
1309. C
1310. C IDIF = NEMAX - NLELEM - NUELEM
1311. C NF = NTMAX - NUETA + 1
1312. C INCR = 0
1313. C DO 2000 I = NF,NTMAX
1314. C INCR = INCR + 1
1315. C LE(NLETA + INCR) = LE(I) - IDIF
1316. C 2000 CONTINUE
1317. C LE(NLETA+1) = NELEM + 1
1318. C RETURN
1319. C END
1320. C SUBROUTINE UPBETA
1321. C
1322. C UPDATES RIGHT-HAND SIDES TO REFLECT NEW BASIS RESULTING FROM
1323. C CURRENT SIMPLEX PIVOT
1324. C SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
1325. C BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1326. C
1327. C IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1328. 1 INTEGER*4 (I-N,Q)
1329. C INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1330. C INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
1331. C DOUBLE PRECISION E(1000)
1332. C REAL A(500)
1333. C
1334. C COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1335. 1 QEO,QBL,QA,QPL,QMI,QZ,QI,QF,CN,QU,NRMAX,NEMAX,QB,QC,
1336. 2 QE,QH,QL,QO,QR,QM,QG,NTMAX
1337. C
1338. C COMMON/BLOCK2/ DEPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
1339. 1 NPIVOT,IPTYPE,RCCST,IFEAS,IPART(122),INCUMB(122),
1340. 2 IVBND(500),IVID(500),IOBND(500)
1341. C

```



```

1220.      CALL SHIFTR(1,3)
1221.      DO 9000 J=1,NCLL
1222.      IF (KINBAS(J)) 8600,8700,9000
1223.      8600 DE = XUB(J)
1224.      GC TO 8750
1225.      8700 DE = XLB(J)
1226.      8750 LL = LA(J)
1227.      KK = LA(J+1) - 1
1228.      DO 8800 I=LL,KK
1229.      IR = IA(I)
1230.      8800 Y(IR) = Y(IR) - A(I)*DE
1231.      9000 CONTINUE
1232.      CALL FTRAN(1)
1233.      CALL SHIFTR(3,2)
1234.      C
1235.      NCFC = NELEM - NETA
1236.      NSTR = NROW - NSLCK
1237.      C
1238.      RETURN
1239.      END
1240.      SUBROUTINE UNPACK(IV)
1241.      C
1242.      C      EXPANDS COMPRESSED MATRIX COLUMN
1243.      C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
1244.      C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1245.      C*****DESCRIPTION OF PARAMETERS*****
1246.      C      IV = PARAMETER INDEXING COLUMN TO BE EXPANDED
1247.      C*****
1248.      C
1249.      IMPLICIT REAL*4 (A,C,E-H,C,P,R-u,Z), REAL*8 (B,D,X,Y),
1250.      1      INTEGER*4 (I-N,Q)
1251.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
1252.      DOUBLE PRECISION E(1000)
1253.      REAL A(500)
1254.      C
1255.      COMMON/BLCK/ ZTOLZE,ZTOLPV,ZTCCST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
1256.      1      QEO,QBL,QA,QPL,QMI,CZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
1257.      2      QE,QH,QL,QO,QR,QM,CG,NTMAX
1258.      C
1259.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,UPROD,
1260.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
1261.      2      IUBJ,IRCHP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
1262.      3      JCULP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
1263.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
1264.      C
1265.      DO 100 I = 1,NROW
1266.      Y(I) = 0.
1267.      100 CONTINUE
1268.      C
1269.      LL = LA(IV)
1270.      KK = LA(IV+1) - 1
1271.      DO 200 I = LL,KK
1272.      IR = IA(I)
1273.      Y(IR) = A(I)
1274.      200 CONTINUE
1275.      C
1276.      RETURN
1277.      END
1278.      SUBROUTINE SHFTL
1279.      C
1280.      C      SUBROUTINE FOR INVERT

```

```

1159.      NELEM = NELEM + 1
1160.      IE(NELEM) = 1
1161.      E(NELEM) = Y(1)
1162.      GC TO 2300
1163.      C
1164.      C          U ETA ELEMENTS
1165.      C
1166.      2200 IE(NELAST) = 1
1167.      E(NELAST) = Y(1)
1168.      NELAST = NELAST - 1
1169.      NUELEM = NUELEM + 1
1170.      2300 CONTINUE
1171.      C
1172.      JH(IRCWP) = IV
1173.      KINBAS(IV) = IRCWP
1174.      NLETA = NLETA + 1
1175.      IE(NELAST) = IRCWP
1176.      IF (J.NE. KR1) GO TO 2330
1177.      E(NELAST) = Y(IRCWP)
1178.      GC TO 2340
1179.      2330 E(NELAST) = 1.
1180.      NETA = NETA + 1
1181.      LE(NETA+1) = NELEM + 1
1182.      2340 NUELEM = NUELEM + 1
1183.      LE(NLAST) = NELAST
1184.      NELAST = NELAST - 1
1185.      NTLAST = NTLAST - 1
1186.      C
1187.      C          UPDATE ROW COUNTS
1188.      C
1189.      DC 2350 I = 1, NROW
1190.      IF (DABS(Y(I)) .LE. ZTOLZE) GO TO 2350
1191.      IF (HREG(I) .GE. 0) GO TO 2350
1192.      HREG(I) = HREG(I) - INCR
1193.      IF (HREG(I) .GE. 0) HREG(I) = -1
1194.      2350 CONTINUE
1195.      HREG(IRCWP) = 0
1196.      3000 CONTINUE
1197.      C
1198.      C          MERGE L AND U ETAS
1199.      C
1200.      3500 NLETA = NETA
1201.      NETA = NLETA + NUETA
1202.      NLELEM = NELEM
1203.      NELEM = NLELEM + NUELEM
1204.      IF (NLELEM .EQ. 0) GO TO 3550
1205.      CALL SHFTC
1206.      C
1207.      C          INSERT SLACKS FOR DELETED COLUMNS
1208.      C
1209.      3550 DC 3600 I = 1, NROW
1210.      IF (JH(I) .NE. 0) GO TO 3600
1211.      JH(I) = 1
1212.      IRCWP = I
1213.      CALL LNPALK(I)
1214.      CALL FTRAN(1)
1215.      CALL WETA
1216.      3600 CONTINUE
1217.      C
1218.      C          UPDATE X
1219.      C

```

```

1093.      NSLCK = NSLCK + 1
1099.      12CC LL = LA(IV)
1100.      KK = LA(IV+1) - 1
1101.      IF (KK .GT. LL) GO TO 1300
1102.      125C IF (ABS(A(LL) - 1.) .LE. 2TOLZE) GO TO 2000
1103.      C
1104.      13CC NUETA = NUETA + 1
1105.      DO 14CC J = LL, KK
1106.      IR = IA(J)
1107.      IF (IR .EQ. 1) GO TO 1390
1108.      IE(NELAST) = IR
1109.      E(NELAST) = A(J)
1110.      NELAST = NELAST + 1
1111.      NUELEM = NUELEM + 1
1112.      GO TO 1400
1113.      135C EP = A(J)
1114.      14CC CONTINUE
1115.      IE(NELAST) = 1
1116.      E(NELAST) = EP
1117.      LE(NLAST) = NELAST
1118.      NELAST = NELAST + 1
1119.      NLAST = NLAST + 1
1120.      NUELEM = NUELEM + 1
1121.      20CC CONTINUE
1122.      205C IF (KRI .EQ. 0) GO TO 3500
1123.      C
1124.      C      DE L-U DECOMPOSITION OF BUMP
1125.      C
1126.      DO 30CC J = LRI, KRI
1127.      IV = VREG(J)
1128.      CALL UNPACK(IV)
1129.      CALL FTRAN(2)
1130.      IRCWP = 0
1131.      IRCMIN = -999999
1132.      DO 21CC I = 1, NRGW
1133.      IF (DABS(Y(I)) .LE. 2TOLPV) GO TO 2100
1134.      IF (HREG(I) .GE. 0) GO TO 2100
1135.      IF (HREG(I) .LE. IRCMIN) GO TO 2100
1136.      IRCMIN = HREG(I)
1137.      IRCWP = I
1138.      21CC CONTINUE
1139.      IF (IRCWP .GT. 0) GO TO 2150
1140.      WRITE(6,8000)
1141.      KINBAS(IV) = 0
1142.      GO TO 3000
1143.      C
1144.      215C INCR = HREG(IRCWP) + 3
1145.      C
1146.      C      WRITE L AND L ETAS
1147.      C
1148.      IF (J .EQ. KRI) GO TO 2160
1149.      NELEM = NELEM + 1
1150.      IE(NELEM) = IRCWP
1151.      E(NELEM) = Y(IRCWP)
1152.      216C DO 23CC I = 1, NRGW
1153.      IF (I .EQ. IRCWP) GO TO 2300
1154.      IF (DABS(Y(I)) .LE. 2TOLZE) GO TO 2300
1155.      IF (HREG(I) .GE. 0) GO TO 2200
1156.      C
1157.      C      L ETA ELEMENTS
1158.      C

```



1037. LL = LA(IV)  
 1038. KK = LA(IV+1) - 1  
 1039. IMCNT = 0  
 1040. DO 1050 I = LL, KK  
 1041. IR = IA(I)  
 1042. IF (HREG(IR) .GE. C) GO TO 1050  
 1043. IMCNT = IMCNT - (HREG(IR) + 1)  
 1044. 1050 CONTINUE  
 1045. MREG(J) = IMCNT  
 1046. 1100 CONTINUE  
 1047. C  
 1048. C SORT COLUMNS INTO MERIT ORDER  
 1049. C USING SHELL SORT  
 1050. C  
 1051. ISD = 1  
 1052. 1106 IF (KRI .LT. 2\*ISD) GO TO 1108  
 1053. ISD = 2\*ISD  
 1054. GO TO 1106  
 1055. 1108 ISD = ISD - 1  
 1056. C END OF INITIALIZATION  
 1057. 1101 IF (ISD .LE. J) GO TO 1107  
 1058. ISK = 1  
 1059. 1102 ISJ = ISK  
 1060. ISL = ISK + ISD  
 1061. ISY = MREG(ISL)  
 1062. ISZ = VREG(ISL)  
 1063. 1103 IF (ISY .LT. MREG(ISJ)) GO TO 1104  
 1064. 1105 ISL = ISJ + ISD  
 1065. MREG(ISL) = ISY  
 1066. VREG(ISL) = ISZ  
 1067. ISK = ISK + 1  
 1068. IF ((ISK + ISD) .LE. KRI) GO TO 1102  
 1069. ISD = (ISD - 1) / 2  
 1070. GO TO 1101  
 1071. 1104 ISL = ISJ + ISD  
 1072. MREG(ISL) = MREG(ISJ)  
 1073. VREG(ISL) = VREG(ISJ)  
 1074. ISJ = ISJ - ISD  
 1075. IF (ISJ .GT. 0) GO TO 1103  
 1076. GO TO 1105  
 1077. 1107 CONTINUE  
 1078. C END OF SORT ROUTINE  
 1079. C  
 1080. C PUT OUT BELOW BUMP ETAS (PART OF U)  
 1081. C  
 1082. 1150 NSLCK = 0  
 1083. NBELCW = 0  
 1084. NELAST = NEMAX  
 1085. NTLAST = NTMAX  
 1086. LE(NTLAST + 1) = NELAST + 1  
 1087. C  
 1088. LR = LR3  
 1089. IF (LR3 .GE. LR4) LR = LR4  
 1090. IF (LR .GT. KR4) GO TO 2050  
 1091. JK = KR4 + 1  
 1092. DO 2000 JJ = LR, KR4  
 1093. JK = JK - 1  
 1094. IV = VREG(JK)  
 1095. I = MREG(JK)  
 1096. NBELCW = NBELCW + 1  
 1097. IF (IV .GT. NRCW) GO TO 1200

```

976.      C
977.      C          PIVCT ABOVE BUMP (PART OF L)
978.      C
979.      NABOVE = NABOVE + 1
980.      IRCWP = IR
981.      CALL UNPACK(IV)
982.      CALL WRETA
983.      NLETA = NETA
984.      JH(IR) = IV
985.      KINBAS(IV) = IR
986.      VREG(J) = VREG(KR1)
987.      KR1 = KR1 - 1
988.      NVREM = NVREM + 1
989.      HREG(IR) = IV
990.      GO TO 940
991.      C
992.      400 IF (HREG(IR) .GE. C) GO TO 800
993.      IRCNT = IRCNT + 1
994.      IRP = IR
995.      800 CONTINUE
996.      C
997.      IF (IRCNT - 1) 810,900,1000
998.      810 WRITE(6,8000)
999.      KINBAS(IV) = 0
1000.      VREG(J) = VREG(KR1)
1001.      NVREM = NVREM + 1
1002.      KR1 = KR1 - 1
1003.      IF (J .GT. KR1) GO TO 1010
1004.      GO TO 320
1005.      C
1006.      C          PUT VECTOR BELOW BUMP
1007.      C
1008.      900 VREG(J) = VREG(KR1)
1009.      NVREM = NVREM + 1
1010.      KR1 = KR1 - 1
1011.      LR3 = LR3 - 1
1012.      VREG(LR3) = IV
1013.      MREG(LR3) = IRP
1014.      HREG(IRP) = 0
1015.      JH(IRP) = IV
1016.      KINBAS(IV) = IRP
1017.      C
1018.      C          CHANGE ROW COUNTS
1019.      C
1020.      940 DO 950 II = LL,KK
1021.      IIR = IA(II)
1022.      IF (HREG(IIR) .GE. C) GO TO 950
1023.      HREG(IIR) = HREG(IIR) + 1
1024.      950 CONTINUE
1025.      IF (J .GT. KR1) GO TO 1010
1026.      GO TO 320
1027.      1000 IF (J .GE. KR1) GO TO 1010
1028.      J = J+1
1029.      GO TO 320
1030.      1010 IF(NVREM .GT. C) GO TO 310
1031.      C
1032.      C          GET MERIT COUNTS
1033.      C
1034.      1020 IF (KR1 .EQ. C) GO TO 1190
1035.      DO 1100 J = LR1,KR1
1036.      IV = VREG(J)

```

```

915.      C
916.      DO 200 I = LR4, KR4
917.      IR = MREG(I)
918.      HREG(IR) = 0
919.      JH(IR) = IR
920.      KINBAS(IR) = IR
921.      200 CONTINUE
922.      C
923.      C          POLL OUT VECTORS BELOW BUMP AND GET ROW COUNTS
924.      C
925.      NBNCNZ = KR4 - LR4 + 1
926.      IF (KR1 .EQ. 0) GO TO 1190
927.      J = LR1
928.      210 IV = VREG(J)
929.      LL = LA(IV)
930.      KK = LA(IV+1) - 1
931.      IRCNT = 0
932.      DO 220 I = LL, KK
933.      NBNCNZ = NBNCNZ + 1
934.      IR = IA(I)
935.      IF (HREG(IR) .GE. 0) GO TO 220
936.      IRCNT = IRCNT + 1
937.      HREG(IR) = HREG(IR) - 1
938.      IRP = IR
939.      220 CONTINUE
940.      IF (IRCNT - 1) 230, 250, 300
941.      230 WRITE(6, 8000)
942.      ECCC FORMAT(16HUMATRIX SINGULAR )
943.      KINBAS(IV) = 0
944.      VREG(J) = VREG(KR1)
945.      KR1 = KR1 - 1
946.      IF (J .GT. KR1) GO TO 310
947.      GO TO 210
948.      C
949.      250 VREG(J) = VREG(KR1)
950.      KR1 = KR1 - 1
951.      LR3 = LR3 - 1
952.      VREG(LR3) = IV
953.      MREG(LR3) = IRP
954.      HREG(IRP) = 0
955.      JH(IRP) = IV
956.      KINBAS(IV) = IRP
957.      IF (J .GT. KR1) GO TO 310
958.      GO TO 210
959.      300 IF (J .GE. KR1) GO TO 310
960.      J = J + 1
961.      GO TO 210
962.      C
963.      C          POLL OUT REMAINING VECTORS ABOVE AND BELOW THE
964.      C          BUMP AND ESTABLISH MERIT COUNTS OF COLUMNS
965.      C
966.      310 NVREM = 0
967.      IF (KR1 .EQ. 0) GO TO 1190
968.      J = LR1
969.      320 IV = VREG(J)
970.      LL = LA(IV)
971.      KK = LA(IV+1) - 1
972.      IRCNT = 0
973.      DO 800 I = LL, KK
974.      IR = IA(I)
975.      IF (HREG(IR) .NE. -2) GO TO 400

```

4110



```

854.      BARRAY(IFN + 1) = BARRAY(IFU + 1)
855.      1000 CONTINUE
856.      RETURN
857.      END
858.      SUBROUTINE INVERT
859.      C
860.      C      COMPUTES INVERSE OF CURRENT BASIS BY LU DECOMPOSITION
861.      C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
862.      C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
863.      C
864.      IMPLICIT REAL*4 (A,C,E-H,G,P,R-W,Z), REAL*8 (B,D,X,Y),
865.      1      INTEGER*4 (I-N,Q)
866.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
867.      DOUBLE PRECISION E(1000)
868.      REAL A(500)
869.      C
870.      COMMON/BLCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRD,QMA,QBA,QFI,
871.      1      QEQ,QBL,QA,QPL,QMI,QZ,QI,QF,CN,QU,NRMAX,NEMAX,QB,QC,
872.      2      QE,QF,QL,QU,QR,QM,QG,NTMAX
873.      C
874.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
875.      1      OY,DE,DF,A,IGNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
876.      2      IUBJ,IRLWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
877.      3      JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
878.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
879.      C
880.      INTEGER*2 MREG,HREG,VREG
881.      DIMENSION MREG(60),HREG(60),VREG(60)
882.      EQUIVALENCE (MREG(1),YTEMP(1)),(HREG(1),YTEMP(31)),(VREG(1),X(1))
883.      C
884.      C      SET PARAMETERS
885.      C
886.      NETA = 0
887.      NLETA = 0
888.      NLETA = 0
889.      NELEM = 0
890.      NLELEM = 0
891.      NUELEM = 0
892.      NAROVE = 0
893.      LE(1) = 1
894.      LR1 = 1
895.      KR1 = 0
896.      LR4 = NROW + 1
897.      KR4 = NROW
898.      C
899.      C      PUT SLACKS AND ARTIFICIALS IN PART 4 AND REST IN PART 1
900.      C
901.      DO 100 I = 1,NROW
902.      IF (JH(I) .GT. NROW) GO TO 50
903.      LR4 = LR4 - 1
904.      MREG(LR4) = JH(I)
905.      VREG(LR4) = JH(I)
906.      GO TO 90
907.      50 KR1 = KR1 + 1
908.      VREG(KR1) = JH(I)
909.      50 HREG(I) = -1
910.      JH(I) = 0
911.      100 CONTINUE
912.      C
913.      KR3 = LR4 - 1
914.      LR3 = LR4

```

```

793.      DOUBLE PRECISION E(1000)
794.      REAL A(500)
795.      C
796.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCCST,ZTOLRJ,ZTCLSM,QRO,QMA,QBA,QFI,
797.      1      QEO,QBL,QA,QPL,QMI,CZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
798.      2      QE,QH,QL,QG,QR,QM,QG,NTMAX
799.      C
800.      COMMON E,XLE(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
801.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
802.      2      IOBJ,IRCWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
803.      3      JCULP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
804.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
805.      C
806.      NELEM = NELEM + 1
807.      IE(NELEM) = IRCWP
808.      E(NELEM) = Y(IRCWP)
809.      C
810.      DO 1000 I = 1,NROW
811.      IF (I.EQ. IRCWP) GO TO 1000
812.      IF (DABS(Y(I)) .LE. ZTOLZE) GO TO 1000
813.      NELEM = NELEM + 1
814.      IE(NELEM) = I
815.      E(NELEM) = Y(I)
816.      1000 CONTINUE
817.      C
818.      NETA = NETA + 1
819.      LE(NETA+1) = NELEM + 1
820.      RETURN
821.      END
822.      SUBROUTINE SHIFTR(ICLD,INEW)
823.      C
824.      C      REARRANGES DATA STORAGE
825.      C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
826.      C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
827.      C*****DESCRIPTION OF PARAMETERS*****
828.      C      ICLD, INEW = PARAMETERS INDICATING STORAGE LOCATIONS FROM AND
829.      C      TO WHICH DATA IS TO BE TRANSFERRED, RESPECTIVELY
830.      C*****
831.      C
832.      IMPLICIT REAL*4 (A,C,E-H,C,P,R-X,Z), REAL*8 (B,D,X,Y),
833.      1      INTEGER*4 (I-N,Q)
834.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
835.      DOUBLE PRECISION E(1000)
836.      REAL A(500)
837.      C
838.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCCST,ZTOLRJ,ZTCLSM,QRO,QMA,QBA,QFI,
839.      1      QEO,QBL,QA,QPL,QMI,CZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
840.      2      QE,QH,QL,QG,QR,QM,QG,NTMAX
841.      C
842.      COMMON E,XLE(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
843.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
844.      2      IOBJ,IRCWP,IVIN,IVOUT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
845.      3      JCULP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
846.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
847.      C
848.      DIMENSION BARRAY(240)
849.      EQUIVALENCE (BARRAY(1),B(1))
850.      IFC = (ICLD - 1) * NRMAX
851.      IFN = (INEW - 1) * NRMAX
852.      C
853.      DO 1000 I = 1,NROW

```

```

732.      100 IF (X(I) .LT. (XLB(ICOL) - ZTOLZE)) GO TO 500
733.      DE = (X(I) - XLB(ICOL))/Y(I)
734.      IF (DE .GE. DP) GO TO 500
735.      IPTYPE = 0
736.      GO TO 250
737.      C      NEGATIVE COEFFICIENT
738.      200 IF (X(I) .GT. (XUB(ICOL) + ZTOLZE)) GO TO 500
739.      DE = (X(I) - XUB(ICOL))/Y(I)
740.      IF (DE .GE. DP) GO TO 500
741.      IPTYPE = -1
742.      250 DP = DE
743.      IRCWP = 1
744.      500 CONTINUE
745.      DE = DP + XLB(JCOLP)
746.      IF (DE .LT. XUB(JCOLP)) GO TO 600
747.      DP = XUB(JCOLP) - XLB(JCOLP)
748.      NPIVCT = 0
749.      RETURN
750.      600 NPIVCT = 1
751.      RETURN
752.      C
753.      C      INCOMING VARIABLE AT UPPER BOUND
754.      C
755.      1000 DP = -1.E10
756.      DO 1500 I=1,NROW
757.      IF (I .EQ. IOBJ) GO TO 1500
758.      ICCL = JH(I)
759.      IF (Y(I) .GT. ZTOLPV) GO TO 1100
760.      IF (Y(I) .LT. -ZTOLPV) GO TO 1200
761.      GO TO 1500
762.      C      POSITIVE COEFFICIENT
763.      1100 IF (X(I) .GT. (XUB(ICOL) + ZTOLZE)) GO TO 1500
764.      DE = (X(I) - XUB(ICOL))/Y(I)
765.      IF (DE .LE. DP) GO TO 1500
766.      IPTYPE = -1
767.      GO TO 1250
768.      C      NEGATIVE COEFFICIENT
769.      1200 IF (X(I) .LT. (XLB(ICOL) - ZTOLZE)) GO TO 1500
770.      DE = (X(I) - XLB(ICOL))/Y(I)
771.      IF (DE .LE. DP) GO TO 1500
772.      IPTYPE = 0
773.      1250 DP = DE
774.      IRCWP = 1
775.      1500 CONTINUE
776.      DE = DP + XUB(JCOLP)
777.      IF (DE .GT. XLB(JCOLP)) GO TO 1600
778.      DP = XLB(JCOLP) - XUB(JCOLP)
779.      NPIVCT = 0
780.      RETURN
781.      1600 NPIVCT = 1
782.      RETURN
783.      END
784.      SUBROUTINE WRETA
785.      C
786.      C      FORMS NEW ETA-VECTOR CORRESPONDING TO CURRENT SIMPLEX PIVOT
787.      C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
788.      C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
789.      C
790.      IMPLICIT REAL*4 (A,C,E-H,I,P,R-W,Z), REAL*8 (B,D,X,Y),
791.      1      INTEGER*4 (I-N,Q)
792.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE

```



```

671.      GC TO 1000
672.      600 IF (DSUM .LE. CMAX) GO TO 1000
673.      CMAX = DSUM
674.      JCCL2 = J
675.      1000 CONTINUE
676.      C
677.      IF (CMIN .LE. -ZTCOST) GO TO 1500
678.      IF (CMAX .GE. ZTCOST) GO TO 2000
679.      JCCLP = J
680.      RCGST = J.
681.      RETURN
682.      1500 IF (CMAX .GE. ZTCOST) GO TO 2500
683.      1600 JCCLP = JCCL1
684.      RCGST = CMIN
685.      RETURN
686.      2000 JCCLP = JCCL2
687.      RCGST = CMAX
688.      RETURN
689.      2500 IF (ABS(CMIN) - CMAX) 2000,2000,1600
690.      END
691.      SUBROUTINE CHJZK
692.      C
693.      C      PERFORMS MIN-RATIO TEST FOR PIVOT COLUMN JCCLP DETERMINED IN
694.      C      SUBROUTINE PRICE. SELECTS PIVOT ROW IROWP FOR CURRENT PRIMAL-
695.      C      SIMPLEX ITERATION.
696.      C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
697.      C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
698.      C
699.      IMPLICIT REAL*4 (A,C,E-H,I,P,K-W,Z), REAL*8 (B,D,X,Y),
700.      1      INTEGER*4 (I-N,Q)
701.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
702.      INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
703.      DCUBLE PRECISION E(1000)
704.      REAL A(500)
705.      C
706.      COMMON/BLCKZ/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTCLSM,QRC,QMA,QBA,QFI,
707.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
708.      2      QE,QF,QG,QC,QK,QM,QG,ATMAX
709.      C
710.      COMMON/BLCKZ/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
711.      1      NPIVLT,IPTYPE,RCGST,IFEAS,IPART(122),INCUMB(122),
712.      2      IVBND(500),IVID(500),ICBND(500)
713.      C
714.      COMMON E,XLB(122),XGB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
715.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
716.      2      IUBJ,IROWP,IVIN,IVOUT,ITCNT,INVERQ,ITRFRQ,ITSINV,IFFEZ,
717.      3      JCCLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NJETA,
718.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
719.      C
720.      IF (KINBAS(JCCLP) .EQ. -1) GO TO 1000
721.      C
722.      C      INCOMING VARIABLE AT LOWER BOUND
723.      C
724.      DP = 1.E10
725.      DO 500 I=1,NROW
726.      IF (I .EQ. ICBND) GO TO 500
727.      ICCL = JH(I)
728.      IF (Y(I) .GT. ZTOLPV) GO TO 100
729.      IF (Y(I) .LT. -ZTOLPV) GO TO 200
730.      GO TO 500
731.      C      POSITIVE COEFFICIENT

```

```

610.      IF (X(I) .GE. (XUB(ICOL) + ZTOLZE)) GO TO 20
611.      Y(I) = 0.
612.      GC TO 30
613.      10 Y(I) = 1.
614.      SUMINF = SUMINF + XUB(ICOL) - X(I)
615.      GC TO 30
616.      20 Y(I) = -1.
617.      SUMINF = SUMINF + X(I) - XUB(ICOL)
618.      30 CONTINUE
619.      IF (SUMINF .GT. ZTOLSM) RETURN
620.      Y(ICBJ) = 1.
621.      IFFEZ = 1
622.      MSTAT = 4F
623.      RETURN
624.      END
625.      SUBROUTINE PRICE
626.      C
627.      C      PRICES OUT NONBASIC COLUMNS, CHOOSES PIVOT COLUMN JCULP FOR
628.      C      CURRENT PRIMAL-SIMPLEX ITERATION
629.      C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
630.      C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
631.      C
632.      IMPLICIT REAL*4 (A,C,E-H,I,P,R-W,Z), REAL*8 (B,D,X,Y),
633.      1      INTEGER*4 (I-N,Q)
634.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
635.      INTEGER*2 IPART,INCUMB,IVBND,IVID,ICBND
636.      DOUBLE PRECISION E(1000)
637.      REAL A(500)
638.      C
639.      COMMON/BLOCK1/ ZTOLZE,ZTOLPV,ZTCGST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
640.      1      QEU,QBL,QA,QPL,QMI,CZ,C1,QF,CN,QU,NRMAX,NEMAX,QB,QC,
641.      2      QE,QF,QL,QD,QK,QM,QG,NTMAX
642.      C
643.      COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
644.      1      NPIVOT,IPTYPE,RCGST,IFEAS,IPART(122),INCUMB(122),
645.      2      IVBND(500),IVID(500),ICBND(500)
646.      C
647.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,UPROD,
648.      1      UY,UE,UP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
649.      2      IOBJ,IRCWP,IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,ITSINV,IFFEZ,
650.      3      JCULP,NRUK,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
651.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
652.      C
653.      CMIN = 1.E10
654.      CMAX = -1.E10
655.      DO 1000 J=1,NCOL
656.      IF (KINBAS(J) .GT. 0) GO TO 1000
657.      IF ((XUB(J) - XLB(J)) .LT. ZTOLZE) GO TO 1000
658.      DSUM = 0.
659.      LL = LA(J)
660.      KK = LA(J+1) - 1
661.      DO 500 I = LL, KK
662.      IR = IA(I)
663.      UE = A(I)
664.      UPROD = UE * Y(IR)
665.      DSUM = DSUM + UPROD
666.      500 CONTINUE
667.      IF (KINBAS(J) .EQ. -1) GO TO 600
668.      IF (DSUM .GE. CMIN) GO TO 1000
669.      CMIN = DSUM
670.      JCCL1 = J

```



```

549.      2      QE,QF,QL,QD,QR,QM,QG,NTMAX
550.      C
551.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
552.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
553.      2      ICBJ,IRCLP,IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,ITSINV,IFFEZ,
554.      3      JCULP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
555.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
556.      C
557.      IF (NETA .LE. 0) GO TO 9000
558.      DO 1000 I = 1,NETA
559.      IK = NETA - I + 1
560.      LL = LE(IK)
561.      KK = LE(IK+1) - 1
562.      IPIV = IE(LL)
563.      DP = E(LL)
564.      DY = Y(IPIV)
565.      DSUM = 0.
566.      IF (KK .LE. LL) GO TO 600
567.      LL = LL + 1
568.      DO 500 J = LL,KK
569.      IR = IE(J)
570.      DE = E(J)
571.      DPRD = DE * Y(IR)
572.      DSUM = DSUM + DPRD
573.      500 CONTINUE
574.      C
575.      600 Y(IPIV) = (DY - DSUM) / DP
576.      1000 CONTINUE
577.      9000 RETURN
578.      END
579.      SUBROUTINE FORMC
580.      C
581.      C      FORMS COST VECTOR FOR CURRENT PRIMAL-SIMPLEX ITERATION; USED IN
582.      C      CONJUNCTION WITH SUBROUTINE PRICE
583.      C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
584.      C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
585.      C
586.      IMPLICIT REAL*4 (A,C,E-H,G,P,R-W,Z), REAL*8 (B,D,X,Y),
587.      1      INTEGER*4 (I-N,Q)
588.      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
589.      DOUBLE PRECISION E(1000)
590.      REAL A(500)
591.      C
592.      COMMON/BLUCK/ ZTOLZE,ZTOLPV,ZTCCST,ZTOLRJ,ZTCLSM,QRO,QMA,QBA,QFI,
593.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,CN,QU,NRMAX,NEMAX,QB,QC,
594.      2      QE,QF,QL,QD,QR,QM,QG,NTMAX
595.      C
596.      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
597.      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
598.      2      ICBJ,IRCLP,IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,ITSINV,IFFEZ,
599.      3      JCULP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
600.      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
601.      C
602.      MSTAT = QI
603.      IFFEZ = J
604.      SUMINF = 0.
605.      Y(ICBJ) = 0.
606.      DO 30 I=1,NROW
607.      IF (I .EQ. ICBJ) GO TO 30
608.      ICOL = JH(I)
609.      IF (X(I) .LE. (XLB(ICOL) - ZTOLZE)) GO TO 10

```



```

488. C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
489. C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
490. C*****DESCRIPTION OF PARAMETERS*****
491. C      IPAR = PARAMETER INDICATING BY WHICH ETA-VECTORS THE MATRIX
492. C      COLUMN IS TO BE UPDATED
493. C*****
494. C
495. C      IMPLICIT REAL*4 (A,C,E-H,O,P,R-h,Z), REAL*8 (B,D,X,Y),
496. C      1      INTEGER*4 (I-N,Q)
497. C      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
498. C      DOUBLE PRECISION E(1000)
499. C      REAL A(500)
500. C
501. C      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
502. C      1      CEG,QBL,QA,QPL,QMI,QZ,QI,QF,CN,QU,NRMAX,NEMAX,QB,QC,
503. C      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
504. C
505. C      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
506. C      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
507. C      2      IOBJ,IRGWP,IVIN,IVOJT,ITCNT,INVFRQ,ITRFRQ,ITSINV,IFFEZ,
508. C      3      JCULP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
509. C      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
510. C
511. C      GO TO (100,110), IPAR
512. C      100 NFE = 1
513. C      NLE = NETA
514. C      GO TO 200
515. C      110 NFE = NLETA + 1
516. C      NLE = NETA
517. C      200 IF (NFE .GT. NLE) GO TO 9000
518. C      DO 1000 IK = NFE,NLE
519. C      LL = LE(IK)
520. C      KK = LE(IK+1) - 1
521. C      IPIV = IE(LL)
522. C      DY = Y(IPIV)
523. C      DY = DY/E(LL)
524. C      Y(IPIV) = DY
525. C      IF (KK .LE. LL) GO TO 1000
526. C      LL = LL + 1
527. C      DO 500 J = LL, KK
528. C      IR = IE(J)
529. C      Y(IR) = Y(IR) - E(J) * DY
530. C      500 CONTINUE
531. C      1000 CONTINUE
532. C      9000 CONTINUE
533. C      RETURN
534. C      END
535. C      SUBROUTINE BTRAN
536. C
537. C      PERFORMS BACKWARD TRANSFORMATION ON COLUMN STORED IN VECTOR Y
538. C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
539. C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
540. C
541. C      IMPLICIT REAL*4 (A,C,E-H,O,P,R-h,Z), REAL*8 (B,D,X,Y),
542. C      1      INTEGER*4 (I-N,Q)
543. C      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
544. C      DOUBLE PRECISION E(1000)
545. C      REAL A(500)
546. C
547. C      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCST,ZTOLRJ,ZTOLSM,CRO,QMA,QBA,QFI,
548. C      1      CEG,QBL,QA,QPL,QMI,QZ,QI,QF,CN,QU,NRMAX,NEMAX,QB,QC,

```

```

427.      ICS1 = NAME(1)
428.      ICS2 = NAME(2)
429.      C
430.      C      TEST FOR ROW MATCH
431.      C
432.      530 DO 540 I = 1,NROW
433.          IF(NAME(J) .NE. ICNAM(I,1) .OR. NAME(K) .NE. ICNAM(I,2))GO TO 540
434.          B(I) = ATEMP1
435.      535 IF(K .GT. 5) GO TO 5
436.          IF(CAES(ATEMP2) .LE. 2TULZE) GO TO 5
437.          J = 5
438.          K = 6
439.          ATEMP1 = ATEMP2
440.          GO TO 530
441.      540 CCNTINCE
442.          WRITE(6,8300) NAME(J),NAME(K)
443.          STCP
444.      C
445.      C      END OF INPUT
446.      C
447.      600 NSCOL = NCOL - NROW
448.      K = NROW + 1
449.      C      INPUT LOWER AND UPPER BOUNDS ON DECISION VARIABLES FOR SUBPROB-
450.      C      LEM INPROB
451.      READ (5,850) (XLB(J), J=K,NCOL)
452.      READ (5,850) (XUB(J), J=K,NCOL)
453.      650 FORMAT (15F5.0)
454.      XLB(ICBJ) = -10000.
455.      XUB(ICBJ) = 10000.
456.      C      INPUT LOWER AND UPPER BOUNDS ON ALLOCATION VARIABLE FOR SUB-
457.      C      PROBLEM INPROB
458.      READ (5,851) (ILB(I,INPROB), I=1,NLINK)
459.      READ (5,851) (IUB(I,INPROB), I=1,NLINK)
460.      651 FORMAT (8I10)
461.      INVFC = NROW
462.      C      STORE A, LA, LA, XUB, OLB, B, NCOL, NROW FOR SUBPROBLEM INPROB
463.      LL = JFIRST(INPROB) - 1
464.      DO 1010 I=1,NELEM
465.          LLPI = LL + I
466.          AMATRS(LLPI) = A(I)
467.      1010 IAS(LLPI) = IA(I)
468.      DO 1020 K=1,NCOL
469.          LAS(K,INPROB) = LA(K)
470.          XLBS(K,INPROB) = XLB(K)
471.      1020 XUBS(K,INPROB) = XUB(K)
472.      K = NLINK + 1
473.      LL = NROW + K
474.      DO 1030 I=1,LL
475.          1030 RPS(I,INPROB) = SNGL(B(K+I))
476.          NCCLS(INPROB) = NCOL
477.          NRCWS(INPROB) = NROW
478.          JFIRST(INPROB+1) = JFIRST(INPROB) + NELEM
479.      C
480.      NELEM = NELEM - NROW
481.      RELEM = NELEM
482.      RDENS = RELEM/(NROW*NCOL)
483.      RETLRN
484.      END
485.      SUBROUTINE FTRAN(IPAK)
486.      C
487.      C      PERFORMS FORWARD TRANSFORMATION ON COLUMN STORED IN VECTOR Y

```



```

366. C      TEST FOR ROW MATCH
367. C
368. 330 DC 340 I = 1,NROW
369.     IF(NAME(J) .NE. ICNAM(I,1) .OR. NAME(K) .NE. ICNAM(I,2))GO TO 340
370.     NELEM = NELEM + 1
371.     IA(NELEM) = I
372.     A(NELEM) = ATEMP1
373.     LA(NCCL+1)=NELEM+1
374. 335 IF(K .GT. 5) GO TO 5
375.     IF(CABS(ATEMP2) .LE. ZTOLZE) GO TO 5
376.     J = 5
377.     K = 6
378.     ATEMP1 = ATEMP2
379.     GO TO 330
380. 340 CONTINUE
381.     WRITE(6,8300) NAME(J),NAME(K),NAME(1),NAME(2)
382. 8300 FORMAT(18HNO MATCH FOR ROW ,2A4,10HAT COLUMN ,2A4)
383.     STOP
384. C
385. C      BASIS CARDS
386. C
387. 410 DC 420 I = 1,NCOL
388.     IF(NAME(1) .NE. ICNAM(I,1) .OR. NAME(2) .NE. ICNAM(I,2)) GO TO 420
389.     IBVEC = I
390.     GO TO 420
391. 420 CONTINUE
392.     WRITE(6,8400) NAME(1),NAME(2)
393. 8400 FORMAT(21HNO MATCH FOR VECTOR ,2A4)
394.     GO TO 5
395. 425 DC 430 I = 1,NROW
396.     IF (NAME(3) .NE. ICNAM(I,1) .OR. NAME(4) .NE. ICNAM(I,2)) GO TO 430
397.     IBRCW=I
398.     GO TO 440
399. 430 CONTINUE
400.     WRITE(6,8500) NAME(3),NAME(4)
401.     GO TO 5
402. 440 JH(IBRCW) = IBVEC
403.     KINEAS(IBRCW) = 0
404.     KINEAS(IBVEC) = IBRCW
405.     GO TO 5
406. C
407. C      RHS
408. C
409. 500 J = 3
410.     K = 4
411.     IF (CABS(ATEMP1) .LE. ZTOLZE) GO TO 521
412.     GO TO 524
413. 521 J=5
414.     K=6
415.     ATEMP1=ATEMP2
416.     GO TO 530
417. 524 CONTINUE
418.     IF (NAME(1) .EQ. ICS1 .AND. NAME(2) .EQ. ICS2) GO TO 530
419. C
420. C      TEST FOR SPLIT VECTOR
421. C
422. DC 525 I = 1,NCOL
423.     IF (NAME(1) .NE. ICNAM(I,1)) GO TO 525
424.     IF (NAME(2) .NE. ICNAM(I,2)) GO TO 525
425.     WRITE(6,8250) NAME(1),NAME(2)
426. 525 CONTINUE

```



```

305.      150 CONTINUE
306.      GO TO 5
307.      C
308.      210 NR0W=NR0W+1
309.      NCCL=NR0W
310.      ICNAM(NR0W,1) = NAME(1)
311.      ICNAM(NR0W,2) = NAME(2)
312.      C
313.      C      TEST ROW TYPE
314.      C
315.      IF(K2.EQ.WL .OR. K3.EQ.WL) GO TO 220
316.      IF(K2.EQ.WE .OR. K3.EQ.WE) GO TO 230
317.      IF(K2.EQ.WC .OR. K3.EQ.WC) GO TO 240
318.      IF(K2.EQ.WN .OR. K3.EQ.WN) GO TO 250
319.      GO TO 250
320.      220 XLB(NR0W) = 0.
321.      XLB(NR0W) = 1.E4
322.      GO TO 250
323.      230 XLB(NR0W) = 0.
324.      XLB(NR0W) = 0.
325.      GO TO 250
326.      240 XLB(NR0W) = 0.
327.      XLB(NR0W) = 1.E4
328.      A(NR0W) = -1.
329.      GO TO 250
330.      250 A(NR0W) = 1.
331.      260 IA(NR0W) = NR0W
332.      LA(NR0W) = NR0W
333.      JH(NR0W) = NR0W
334.      KINEAS(NR0W) = NR0W
335.      NELEM=NR0W
336.      GO TO 5
337.      C
338.      C      MATRIX ELEMENTS
339.      C
340.      320 J = 3
341.      K = 4
342.      IF (CABS(ATEMP1) .LE. ZTOLZE) GO TO 321
343.      GO TO 324
344.      321 J=5
345.      K=6
346.      ATEMP1=ATEMP2
347.      GO TO 330
348.      324 CONTINUE
349.      IF (NAME(1) .EQ. ICS1 .AND. NAME(2) .EQ. ICS2) GO TO 350
350.      C
351.      C      TEST FOR SPLIT VECTOR
352.      C
353.      DO 325 I = 1,NCCL
354.      IF (NAME(1) .NE. ICNAM(I,1)) GO TO 325
355.      IF (NAME(2) .NE. ICNAM(I,2)) GO TO 325
356.      WRITE(6,8250) NAME(1),NAME(2)
357.      8250 FORMAT(14HUSPLIT VECTOR ,2A4)
358.      325 CONTINUE
359.      NCCL = NCCL + 1
360.      ICS1 = NAME(1)
361.      ICS2 = NAME(2)
362.      ICNAM(NCCL,1) = ICS1
363.      ICNAM(NCCL,2) = ICS2
364.      LA(NCCL) = NELEM + 1
365.      C

```

```

244. C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
245. C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
246. C*****DESCRIPTION OF PARAMETERS*****
247. C      IFFRUB = INPRUB = IDENTIFICATION NUMBER OF SUBPROBLEM BEING
248. C      INPUT (USER MUST INPUT SUBPROBLEMS IN SEQUENCE 1,2,...,NSUBS)
249. C*****
250. C
251. C      IMPLICIT REAL*4 (A,C,E-H,G,P,K-W,Z), REAL*8 (B,D,X,Y),
252. C      1      INTEGER*4 (I-N,Q)
253. C      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
254. C      INTEGER*2 IZSTAR,IEID,IBBND,IZBND
255. C      INTEGER*2 IAS,LAS
256. C      DOUBLE PRECISION E(1000),ATEMP1,ATEMP2
257. C      REAL A(500)
258. C
259. C      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCCST,ZTOLRJ,ZTCLSM,QRO,QMA,QBA,QF1,
260. C      1      QEC,CBL,QA,QPL,QM1,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
261. C      2      QE,QF,QL,QG,QR,QM,QG,NTMAX
262. C
263. C      COMMON/BLOCK3/ KLAMDA(5,20,10),ZNAUT(20,10),ZJBAR(10),NLINK,NSUBS,
264. C      1      IZINC,IZBAR,IDUM,IBLL,NJFIX,JFIXSM,IB(5),JFIX(10),NSOL(10),
265. C      2      NLAMDA(10),IBSTAR(5,40,10),IBUSE(5,40,10),IBUB(5,10),IBLB(5,10),
266. C      3      IZSTAR(40,10),IBID(2,200),IBBND(200),IZBND(200)
267. C      COMMON/BLOCK4/ XLBS(20,10),XUBS(20,10),AMATRS(1000),RHS(10,10),
268. C      1      INRCWS(10),NCOLS(10),JFIRST(11),IAS(1000),LAS(20,10)
269. C      COMMON E,XLB(122),XCB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
270. C      1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
271. C      2      IOBJ,IRCWP,IVIN,IVOUT,ITCNT,INVRQ,ITRFRQ,ITSINV,IFFEZ,
272. C      3      JCULP,NROW,NCOL,NELEM,NCTA,NLELEM,NLETA,NUELEM,NUETA,
273. C      4      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
274. C
275. C      DO 10 I=1,60
276. C      10 B(I) = 0.
277. C      DO 20 J=1,122
278. C      20 KINBAS(J) = 0
279. C      NROW = 0
280. C      ITCNT = 0
281. C      ICS1=C
282. C      ICS2=C
283. C      READ (5,7000) INPRUB
284. C      7000 FORMAT (5I4)
285. C      IFPRUB = INPRUB
286. C      IF (IFFRUB .EQ. 0) RETURN
287. C
288. C      5 READ(5,101) K1,K2,K3,K4,(NAME(I),I=1,4),ATEMP1,NAME(5),NAME(6),
289. C      1ATEMP2
290. C      101 FORMAT(4A1,2A4,2X,2A4,2X,F12.4,3X,2A4,2X,F12.4)
291. C      IF(K1 .EQ. QE) GO TO 600
292. C      IF(K1 .EQ. QBL) GO TO 50
293. C      IF(K1 .EQ. QN) GO TO 100
294. C      IF(K1 .EQ. QR .AND. K2 .EQ. QO) L=1
295. C      IF(K1 .EQ. QR .AND. K2 .EQ. QO) GO TO 150
296. C      IF(K1 .EQ. QC) L=2
297. C      IF(K1 .EQ. QC) GO TO 150
298. C      IF(K1 .EQ. CB .AND. K2 .EQ. QA) L=3
299. C      IF(K1 .EQ. CB .AND. K2 .EQ. QA) GO TO 150
300. C      IF(K1 .EQ. CR .AND. K2 .EQ. QH) L=4
301. C      IF(K1 .EQ. CR .AND. K2 .EQ. QH) GO TO 150
302. C      50 GO TO(210,320,410,500),L
303. C      100 NTEMP(1) = NAME(3)
304. C      NTEMP(2) = NAME(4)

```

```

183.      331C AV(J)=((NTSOL(J)-1)*AV(J)+(ZJBAR(J)-INCVL))/NTSOL(J)
184.      GO TO 3400
185.      332C ISLACK = 0
186.      NP1 = NLINK + 1
187.      DC 3325 I=2,NP1
188.      3325 ISLACK = ISLACK + INCUMB(I)
189.      AV(J)=((NTSOL(J)-1)*AV(J)+ISLACK)/NTSOL(J)
190.      GO TO 3400
191.      333C AV(J)=((NTSOL(J)-1)*AV(J)-TIME)/NTSOL(J)
192.      C      SAVE ALLOCATION AND SOLUTION INFORMATION FOR SUBPROBLEM J
193.      340C CALL SAVER(J)
194.      K = NTEMP(2)
195.      C      OUTPUT SOLUTION INFORMATION FOR SUBPROBLEM J
196.      WRITE (6,2) J,K,(INCUMB(KK), KK=1,NCOL)
197.      2 FORMAT (' SUB',12,' SOL',13,' : ',2CI5)
198.      C      TEST IF ALL SUBPROBLEMS HAVE BEEN SOLVED; IF SO, NO BRANCHING
199.      C      NECESSARY
200.      IF (NJFIX - NSUBS + 1) 2200,2300,9959
201.      C      '
202.      C      COMPUTATIONS COMPLETED; OUTPUT OPTIMAL SOLUTION INFORMATION
203.      500C CALL CUTSOL(INC)
204.      CALL FCLOCK(JTIME, ITCT)
205.      TTCT = JTIME/100.
206.      C      OUTPUT ALGORITHMIC PERFORMANCE INFORMATION
207.      WRITE (6,7) TTCT
208.      7 FORMAT (' TOTAL SOLUTION TIME =',F7.2,' SECONDS')
209.      WRITE (6,92) TPH1,NALG,TALD,(NTSOL(J), J=1,NSUBS)
210.      92 FORMAT (' TIME IN PHASE1 =',F6.2,' SECONDS') MASTER PROBLEM SOLVE
211.      10',11C,' TIMES: TOTAL TIME SOLVING MASTER PROBLEM =',F6.2,' SECON
212.      20S') NUMBER OF TIMES EACH SUBPROBLEM SOLVED =',18X,10I5)
213.      WRITE (6,9) (NSOL(J), J=1,NSUBS)
214.      9 FORMAT (' NUMBER OF DISTINCT SOLUTIONS OBTAINED IN EACH SUBPROBLEM
215.      1 =',1CI5)
216.      WRITE (6,91) TSUBS
217.      91 FORMAT (' TOTAL TIME SOLVING SUBPROBLEMS =',F6.2,' SECONDS')
218.      GO TO 10
219.      9959 STOP
220.      END
221.      BLOCK DATA
222.      C
223.      C      INITIALIZES GLOBAL PROGRAM CONSTANTS
224.      C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
225.      C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
226.      C
227.      IMPLICIT REAL*4 (A,C,E-H,G,P,R-W,Z), REAL*8 (B,D,X,Y),
228.      1      INTEGER*4 (I-N,Q)
229.      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLRJ,ZTOLSM,QRO,QMA,QBA,QFI,
230.      1      QEO,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
231.      2      QE,QH,QL,QO,QR,QM,QG,NTMAX
232.      REAL ZTOLZE/1.E-5/,ZTOLPV/1.E-4/,ZTCOST/1.E-3/,ZTOLRJ/1.E-4/,
233.      1      ZTOLSM/1.E-10/
234.      INTEGER*4 NRMAX/60/,NTMAX/250/,NEMAX/1000/
235.      INTEGER*4 QRO/'ROW' //,QMA/'MATR'//,QEA/'BAS1'//,QFI/'FIRS'//,
236.      1QEC/'EDF' //,QBL/' //,QA/'A' //,QPL/' + ' //,QMI/' - ' //,
237.      2QZ/' 2 ' //,CI/'I' //,QF/'F' //,QN/'N' //,CU/'U' //,
238.      3QB/'B' //,QC/'C' //,QE/'E' //,QH/'H' //,QL/'L' //,
239.      4CC/'C' //,CR/'R' //,QM/'M' //,QG/'G' //
240.      END
241.      SUBROUTINE INPUT(1FPROB)
242.      C
243.      C      INPUTS PROBLEM DATA

```



```

122. C      CHECK IF ANY SUBPROBLEM HAS PREVIOUSLY BEEN SOLVED UNDER ITS
123. C      CURRENT ALLOCATION
124. CALL COMCHK(J,K)
125. IF (ICCM .EQ. 0) GO TO 3000
126. C      BRANCH ON ALLOCATION VARIABLE IN SUBPROBLEM J
127. 2200 CALL CMBRAN(J,K)
128. IF (MSTAT .EQ. 0) GO TO 2000
129. C      FIX ALLOCATION TO SUBPROBLEM J AT LEVEL K
130. 2300 JFIX(J) = K
131. NJFIX = NJFIX + 1
132. JFIXSM = JFIXSM + IZSTAR(K,J)
133. IF (NJFIX .LT. NSUBS) GO TO 2100
134. C      ALL SUBS FIXED: COMPARE SOLUTION WITH INCUMBENT
135. IF (JFIXSM .LE. IZINC) GO TO 2500
136. DO 2400 I=1,NSUBS
137. 2400 INC(I) = JFIX(I)
138. IZINC = JFIXSM
139. CALL FCLOCK(JTIME,ITCT)
140. TCPT = JTIME/100.
141. WRITE (6,93) TCPT,IZINC
142. 93 FORMAT (' TIME =',F7.2,' SECONDS: IZINC =',I10)
143. 2500 JFIX(J) = 0
144. JFIXSM = JFIXSM - IZSTAR(K,J)
145. NJFIX = NJFIX - 1
146. GO TO 2000
147. C
148. C      SELECT NEXT SUBPROBLEM TO SOLVE
149. 3000 AVMAX = -10000.
150. J = 0
151. DO 3100 I=1,NSUBS
152. IF (JFIX(I) .GT. 0) GO TO 3100
153. IF (AVMAX .GE. AV(I)) GO TO 3100
154. J = I
155. AVMAX = AV(I)
156. 3100 CONTINUE
157. 3200 NTSOL(J) = NTSOL(J) + 1
158. CALL FCLOCK(JTIME,ITCT)
159. C      LOAD PROBLEM DATA FOR SUBPROBLEM J
160. CALL LDDATA(J)
161. IPAR = 0
162. ITSIN = 99999
163. IFEAS = 0
164. C      GET OPTIMAL LP-SOLUTION TO SUBPROBLEM J
165. CALL NCRML
166. C      TEST IF SUBPROBLEM J IS LP-FEASIBLE
167. IF (MSTAT .EQ. 0) GO TO 3300
168. C      STORE OPTIMAL DUAL MULTIPLIERS IF DISTINCT FROM PREVIOUS MULTS.
169. CALL FISAVR(J)
170. C      SUBPROBLEM J LP-FEASIBLE; GET INITIAL LOWER BOUND ESTIMATE ON
171. C      MAX. IP-CBJ. VAL.
172. CALL GETLBC(J)
173. INITBC = INCUMB(I)
174. C      USE BRANCH-AND-BOUND ROUTINE TO COMPUTE OPT. IP-SOLUTION TO
175. C      SUBPROBLEM J
176. CALL EANDB(INITBC)
177. 3300 CALL FCLOCK(JTIME,ITOT)
178. TIME = (JTIME1 - JTIME)/100.
179. TSUBS = TSUBS + TIME
180. C      UPDATE SUBPROBLEM SELECTION RULE INFORMATION FOR SUBPROBLEM J,
181. C      IN ACCORDANCE WITH VALUE SPECIFIED FOR IPARM
182. GO TO (3310,3320,3330,3400),IPARM

```

```

61.      C
62.      C          PHASE 1
63.      C
64.      DO 100 J=1,NSUBS
65.      NLAMDA(J) = 0
66.      NSCL(J) = 0
67.      AV(J) = 0.
68.      NSCL(J) = 1
69.      JFIX(J) = 0
70.      CALL INPUT(J)
71.      DO 50 I=1,NLINK
72.      50 IBSTAR(I,1,J) = IDINT(B(I+1))
73.      ITSINV = 99999
74.      CALL NCRMAL
75.      100 CALL FISAVR(J)
76.      C
77.      200 CALL ALLOCTE(IPAR)
78.      NNEWPI = 0
79.      DO 300 J=1,NSUBS
80.      LL = NSCL(J)
81.      LISTL = LL + 1
82.      DO 250 K=1,LL
83.      DO 240 I=1,NLINK
84.      IF (IBSTAR(I,K,J) .NE. IBSTAR(I,LISTL,J)) GO TO 250
85.      240 CONTINUE
86.      GO TO 300
87.      250 CONTINUE
88.      CALL LDATA(J)
89.      ITSINV = 99999
90.      CALL NCRMAL
91.      NEWPI = NLAMDA(J)
92.      CALL FISAVR(J)
93.      NEWPI = NLAMDA(J) - NEWPI
94.      NNEWPI = NNEWPI + NEWPI
95.      NSCL(J) = LISTL
96.      300 CONTINUE
97.      IF (NNEWPI) 400,400,200
98.      400 DO 500 J=1,NSUBS
99.      500 NSCL(J) = 0
100.     CALL FCLOCK(JTIME,ITOT)
101.     TPH1 = JTIME/100.
102.     GO TO 2100
103.     C
104.     C          PHASE 2
105.     C
106.     C          BACKTRACK TO LAST PROMISING NODE IN ALLOCATION-VARIABLES SEARCH
107.     C          TREE
108.     2000 CALL EBKTRK
109.     C          IF LIST OF STORED NODES IS EMPTY, COMPUTATIONS ARE COMPLETED
110.     IF (IELL .EQ. 0) GO TO 9000
111.     2100 NALC = NALC + 1
112.     CALL FCLOCK(JTIME,ITOT)
113.     C          SOLVE MASTER PROBLEM TO OBTAIN PROMISING NEW ALLOCATION OF
114.     C          LINKING RESOURCE TO SUBPROBLEMS
115.     CALL ALLOCTE(IPAR)
116.     CALL FCLOCK(JTIME1,ITOT)
117.     TALC = TALC + (JTIME1 - JTIME)/100.
118.     IPAR = 1
119.     C          TEST FOR FATHOMING OF CURRENT NODE
120.     IF (INCVAL .LE. IZINC) GO TO 2000
121.     IZBAR = INCVAL

```

```

1.      C
2.      C      DECOMPOSITION ROUTINE, DMDC
3.      C      DECOMPOSITION AND SOLUTION OF PURE-INTEGER LINEAR PROGRAMS
4.      C      (WITH GENERAL INTEGER VARIABLES) WITH BLOCK ANGULAR CONSTRAINT
5.      C      MATRICES.
6.      C*****MULTIPLE LINKING CONSTRAINTS CASE*****
7.      C      GARY A. KUCHMAN (OPERATIONS RESEARCH), SEPT. 1976, MOD 1
8.      C
9.      C      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
10.     1      INTEGER*4 (I-N,Q)
11.     1      INTEGER*2 JH,KINBAS,LA,LE,IA,IE
12.     1      INTEGER*2 IPART,INCLMB,IVBND,IVID,ICBND
13.     1      INTEGER*2 IZSTAR,IBID,IBBND,IZBND
14.     1      INTEGER*2 IAS,LAS
15.     1      DOUBLE PRECISION E(1000)
16.     1      DIMENSION A(500),INC(10),AV(10),NTSCL(10)
17.     1      COMMON/BLOCK1/ ZTCLZE,ZTCLPV,ZTCLST,ZTCLRJ,ZTCLSM,QRO,QMA,QBA,QFI,
18.     1      QEU,QBL,QA,QPL,QMI,QZ,QI,QF,QN,QU,NRMAX,NEMAX,QB,QC,
19.     1      QG,QF,QL,QQ,QR,QM,QG,NTMAX
20.     1      COMMON/BLOCK2/ DFPART(122),REVBND,INCVAL,ICOL,LISTL,IVAL,IDIR,
21.     1      NPVCT,IPTYPE,RCGST,IFEAS,IPART(122),INCUMB(122),
22.     1      IVBND(500),IVID(500),ICBND(500)
23.     1      COMMON/BLOCK3/ RLAMDA(5,20,10),ZNAUT(20,10),ZJBAR(10),NLINK,NSUBS,
24.     1      IIZINC,IZBAR,IDCM,IBLL,NJFIX,JFIXSM,IB(5),JFIX(10),NSOL(10),
25.     1      ZNLAMDA(10),IBSTAR(5,40,10),IBUSED(5,40,10),IBUB(5,10),IBLB(5,10),
26.     1      IIZSTAR(40,10),IBID(2,200),IBBND(200),IZBND(200)
27.     1      COMMON/BLOCK4/ XLBS(20,10),XUBS(20,10),AMATRS(1000),RHS(10,10),
28.     1      INROWS(10),NCOLS(10),JFIRST(11),IAS(1000),LAS(20,10)
29.     1      COMMON E,XLB(122),XUB(122),B(60),X(60),Y(60),YTEMP(60),DSUM,DPROD,
30.     1      DY,DE,DP,A,ICNAM(122,2),NAME(20),NTEMP(20),SUMINF,MSTAT,
31.     1      IOBJ,IROWP,IVIN,IVDUT,ITCNT,INVERQ,ITRFRQ,ITSINV,IFFEZ,
32.     1      JCULP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,NUELEM,NUETA,
33.     1      IE(1000),IA(500),LE(252),LA(122),KINBAS(122),JH(60)
34.     C
35.     C      MAIN PROGRAM
36.     C      COORDINATES APPLICATION OF THE VARIOUS STEPS OF THE DECOMPOSITION
37.     C      ALGORITHM.
38.     C
39.     C      INITIALIZE PARAMETERS
40.     10 CALL FCLOCK(ITCT)
41.     1 READ(5,1,END=9999) NSUBS,NLINK,IPARM,IZINC
42.     1 FORMAT(5I10)
43.     1 READ(5,1) (IB(I), I=1,NLINK)
44.     1 DO 20 J=1,122
45.     1     ICNAM(J,1) = 0
46.     20 ICNAM(J,2) = 0
47.     1 ICBJ = 1
48.     1 ITRFRQ = 99999
49.     1 INITBE = 0
50.     1 IBLL = 0
51.     1 IZBAR = 10000
52.     1 JFIX(NSUBS) = 0
53.     1 IPAR = 0
54.     1 NJFIX = 0
55.     1 JFIXSM = 0
56.     1 JFIRST(1) = 1
57.     1 NALC = 0
58.     1 TALC = 0
59.     1 TSUBS = 0
60.     1 NPI = NLINK + 1

```



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 71	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTER PROGRAMS FOR DECOMPOSITION IN INTEGER PROGRAMMING		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Gary A. Kochman		8. CONTRACT OR GRANT NUMBER(s) N00014-76-C-0418
9. PERFORMING ORGANIZATION NAME AND ADDRESS Dept. of Operations Research Stanford University Stanford, California		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (NR-047-061)
11. CONTROLLING OFFICE NAME AND ADDRESS Logistics & Mathematical Statistics Branch Code 434 Office of Naval Research Arlington, Va. 22217		12. REPORT DATE September 1976
		13. NUMBER OF PAGES 162
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release, distribution is unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES This research was supported in part by: U.S. Army Research Office Contract DAAC29-74-C-0034 and Air Force Office of Scientific Research Contract F44620-74-C-0079.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Mathematical Programming, Integer Programming, Decomposition		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report gives documentations for two computer codes, DSLC and DMLC for solving block angular integer programming problems. The first code, DSLC, is for the single linking constraint case and the second DMLC is for the multiple linking constraint case.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE  
S/N 0102-014-6601UNCLASSIFIED  
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)